UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

TRABAJO DE FIN DE GRADO

Aplicación de técnicas de Data Analytics para la evaluación y mejora de la calidad en comunicaciones digitales

Departamento de Ingeniería Telemática

Grado en Ingeniería en Tecnologías de Telecomunicación

Autor: ORLANDO MORA GARCÍA

Tutor: JULIO VILLENA ROMÁN

Agradecimientos

Me gustaría agradecer al profesor Julio Villena el haberme guiado durante la elaboración de este proyecto, su disposición para atenderme y los buenos consejos que he recibido.

También agradecer a la universidad pública en general y a la Universidad Carlos III en particular por haberme permitido formarme y por las amistades que han surgido en ella, que confío en que durarán toda la vida.

Quiero destacar la labor de mis padres dando tanto valor a la formación de mis hermanos y mía, dedicando todos sus esfuerzos para ello y enseñándonos una gran lección.

Además de ellos, agradezco a mis abuelos y a mi tía Angelita el inculcarme la humildad, la cultura del esfuerzo y la alegría.

Por último, a mi compañera por permitirme crecer a su lado etapa tras etapa.

Índice

Índice de figuras

Índice de tablas

Glosario

ADSL: Asymetric Digital Subscriber Line

BRAS: Broadband Remote Access Server

CPE: Customer Premise Equipment

CRISP-DM: Cross Industry Standard Process for Data Mining

CSV: Comma separated values

DEWP: Mean dew point

DSLAM: Digital Subscriber Line Access Multiplexer

FRSHTT: Fog, Rain, Snow, Hail, Thunder and Tornado

IP: Internet Protocol

LOPD: Ley Orgánica de Protección de Datos

LOPGDD: Ley Orgánica de Protección de Datos y Garantía de Derechos Digitales

MXSPD: Maximum sustained wind speed

NAT: Network Adress Translation

NCDC: National Climatic Data Center

PCA: Principal Componet Analysis

PRCP: Total precipitation

RGPD: Reglamento General de Protección de Datos

SLP: Sea Level Pressure

SNDP: Snow Depth

STP: Station Pressure

SVM: Support Vector Machine

TFG: Trabajo de fin de grado

VDSL: Very high-bit-rate Digital Subscriber Line

WBAN: Weather Buerau Air Force Navy

WDSP: Wind Speed

Extended abstract

Nowadays the world cannot be understood without telecommunications. The instantaneous transmission of information has changed the way we understand society. Personal relationships, business models or consumer habits are just a few examples of everyday actions that have been affected by the arrival of the Internet in people's lives.

This End of Degree Paper (TFG) focuses on predicting connection quality degradation on ADSL and VDSL lines taking into account meteorological data from the different cities for which information is available, due to the suspicion that different atmospheric events may have a negative impact on the quality of communications.

The objective of this project has two applications. In the corporative aspect, the idea is to try to satisfy a business case for the different network providers that consists in the reduction of costs for calls of customers to the technical service and technical dispatches in the customer's home. In the personal area, it aims to acquire knowledge throughout the world of data science, including the different analytical data libraries for Python language or SQL language to obtain the necessary information from the database.

It should not be forgotten that a very important part of any project of this kind must be based on an understanding of the subject that will be analyzed. The experience of close to two years working in a company in the sector has allowed the project developer to know the detail of the procedures of the diagnostics that are performed and the commercial relevance that can have each of the situations that can cause a bad line performance.

Carrying out a project of data analysis from the beginning to the end will help to get fluency with the tools used, to know all the difficulties that may arise and to internalize good practices during development.

Motivation

As mentioned earlier, Internet connectivity has changed society in recent years. This has led to multiple business opportunities, especially for operating companies, and has made the sector one of the most profitable.

The motivation of this project can be divided into three applications:

- At the business level: For companies that provide communications services, a good quality service is essential to maintain their business model.

  The two most important aspects in economic terms are customer churn and interventions in the location of the installation. If, after the development of the project, it were possible to predict more accurately the

instability in the customer lines, the possibility of carrying out preventive interventions in the field would appear.

In this way, incidences will be resolved before the customer detects a degradation in the service, reducing the expense derived from calls to customer service and avoiding redundancy in interventions in case of massive failure, at the same time as improving the corporate image.

- On a personal level: The choice to study the Degree in Engineering in Telecommunication Technologies in 2012 already brought with it the motivation of being part of a sector such as telecommunications, at the leading edge of technology and with a strong day-to-day presence in society.

  Added to this interest is the attraction of a sector as important as data science. The aim of this work is to carry out a data analysis project from beginning to end and, with it, familiarisation with the techniques and technologies to be used, as well as the steps to be followed and the good practices to be taken.

- At a social level: Given the importance of activities that require an Internet connection at a social, cultural, educational or administrative level, among other examples, there may be situations of social inequality when not being able to afford a quality Internet connection negatively affects equal opportunities.

  For this reason, the improvement in services at network level also seeks to provide society with digital resources that can be used as tools for social cohesion. An example of this could be distance education or health care in rural areas.

Goals

The business of network companies is based on the sale of phone and Internet services to both companies and private individuals.

The revenue these companies receive is directly related to the number of customers and the price of the products they sell. Expenses are more diverse, ranging from the establishment of new networks, renewal and troubleshooting in networks already implemented to customer service.

The business case pursued in this project aims to maximize revenue and reduce costs. For this reason, the main problem to be solved is the poor quality of the connection that customers can perceive.

Poor data line performance means, on the one hand, a reduction in a company's revenue by increasing the risk of customer churn and a degradation in corporate reputation by not satisfying the needs for which the services are ordered.

In addition, the costs related to customer service increase, especially the increase in calls to customer service and a greater number of issues to be resolved in the place where they occur.

This specific project focuses on studying the relationship that may exist between the meteorological events that occur in a city and the quality of the lines in that same city.

The model to be developed during this investigation would make it possible to know in advance the probabilities of an increase in incidents in a specific area based on meteorological information.

With this, a proactive policy of incident resolution could be carried out, before the customer perceives instability in the connection, instead of reactive, taking action when the customer reports the situation.

Memory contents

This report is organized to be as understandable as possible. On the other hand, it is intended to follow the structure of a machine learning project, explained later in section 2.6.

Section 2 responds to the first step of the process which is the understanding of the technology you want to work with.

Sections 3 and 4 refer to the available data understanding, both line performance and meteorological data.

In Section 5, the data are processed to make them suitable for the creation of the models shown in Section 6.

Chapter 8 evaluates the results obtained and chapter 9 defines the further steps to be taken for improving the analysis.

Schedule

This section details the timeline followed in carrying out the project.

ID	Start Date	End Date	Task	Time in days
1	02/03/2019	03/03/2019	ADSL Networks	2
2	02/03/2019	03/03/2019	Machine Learning	2
3	03/03/2019	06/03/2019	Data modeling	4
4	07/03/2019	07/03/2019	Hypothesis contrast	1
5	08/03/2019	12/03/2019	Use case establishment	5
6	13/03/2019	18/03/2019	Available performance data inventory	6
7	19/03/2019	23/03/2019	Meteorological data search	5
8	24/03/2019	07/04/2019	Collection and processing of performance data	15
9	08/04/2019	22/04/2019	Meteorological data collection and processing	15
10	23/04/2019	02/05/2019	Applying Different Models to the Data Set	10
11	03/05/2019	12/05/2019	Conclusions and next steps	10
12	13/05/2019	27/05/2019	Memory writing	15

Tabla 0-1: Project schedule

Next, Gantt Diagram is shown:

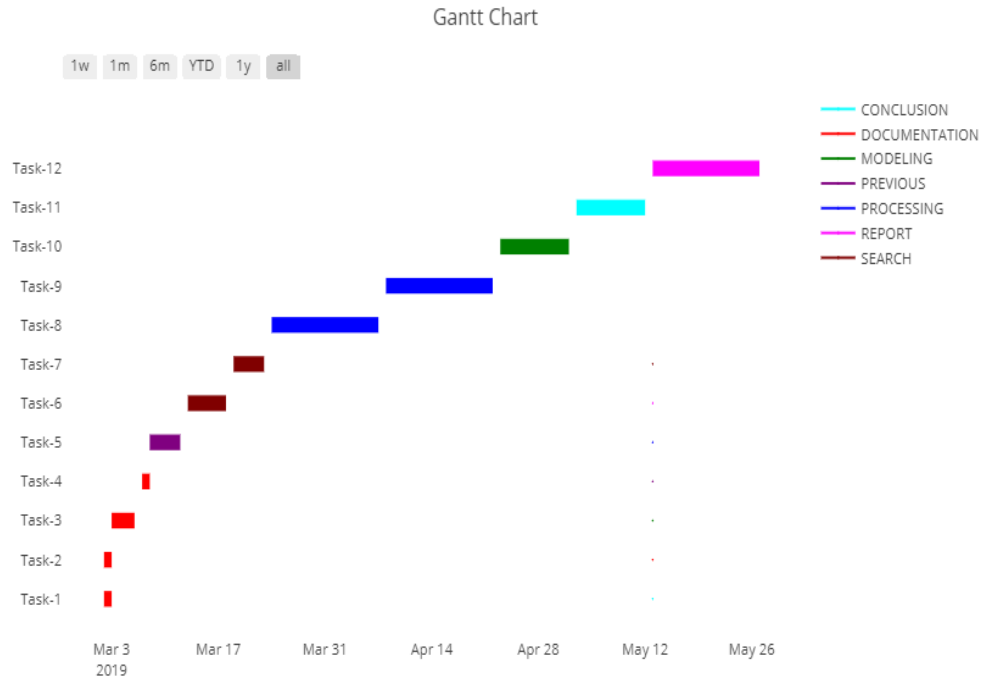

Figura 0-1: Gantt diagram

Budget

This section shows the budget required to perform the project. This includes both the staff cost and material costs.

Firstly, as mentioned in the schedule section, 90 working days over 18 weeks are required. It is also estimated that 4 hours a day will be dedicated to the project.

In addition, it is necessary to take into account the hours spent by the tutor for the guidance and correction of errors that have taken two hours per week throughout the duration of the project.

Staff	Working hours (h)	Cost per hour (€/h)	Cost (€)
Tutor	36	80	2880
Student	360	40	14400
		TOTAL	17280

Tabla 0-2: Staff costs

To the personal costs it is also necessary to add the indirect costs, corresponding to 10% of these: 1,728 €.

On the other hand, it is necessary to take into account the costs associated with the material used. Given that both Python and the libraries used are open source, there have not been costs associated with software. However, it has been necessary to acquire a laptop with enough capacity to process a large amount of data and a SSD hard drive to optimize the performance of the equipment.

Depreciation must also be taken into account when calculating the material cost. The depreciation coefficient established by the Spanish Tax Agency [1] is applied. The formula for calculating the cost would be as follows:

Cost = Depreciation coefficient x No. of months of usage / 12 x Material Price

Material	Unit cost (€)	Usage (months)	Deprecation coef.	Cost (€)
Laptop	600	4	0,26	52,00
SSD Hard Drive	150	4	0,26	13,00
			TOTAL	65,00

Tabla 0-3: Material costs

Budget	
Material costs	65
Staff costs	17280
Indirect costs	1728
Total	19073
VAT (21%)	4005,33
TOTAL (VAT incl.)	23078,33

Tabla 0-4: Total costs

By adding material and personal costs, the total cost amounts to 23,078.33 €.

Conclusions

In order to discuss the conclusions of this work it is necessary to look at the goals of this work. These goals can be divided into two planes:

- Business area: The motivation for this project was to carry out an exploration of how climatology affects the performance of the lines and what the process would be like to try to obtain measurable results from this phenomenon.

  While it is true that the conclusion that can be drawn from this project is that there is no measurable relationship between meteorological data and the quality of communications, it can be said that it can be the beginning of a deeper data analysis project.

  Following the steps of the CRISP-DM process diagram, we are now in the evaluation phase of the model, returning to the point of business and data knowledge. Given that the current process does not obtain the expected result, it is necessary to redefine the business case or carry out a process of searching, obtaining and processing new data that are useful for the creation of new models.

- Academic area: Despite the fact that the results of the analyses have not been as expected, from an educational point of view this project has been successful.

  Carrying out a data analysis project has required the preparation of an use case, which involves understanding the business area to be dealt with. In addition, obtaining data and understanding the available information has required redefining the project.

  From the technical point of view, this work has helped to understand and get familiar with the most commonly used tools in this kind of projects, especially the SQL language and the Python language with its libraries pandas, seaborn or scikit-learn.

Further Steps

As mentioned in the previous section, the end of this project does not suppose the end of the analysis. There are multiple action points to obtain more suitable data for this analysis, as well as a possible redefinition of the bussines case.

First of all, the first thing to do would be to increase the available amount of data. The current dataset contains data for three months. It would be interesting to study how the lines behave in different stations and, although a greater amount of data does not necessarily mean a greater amount of useful information, it is very likely that the results obtained will be more precise.

Another point that would help to obtain a more accurate result would be not to group the data by city and day since the number of samples is considerably decreased. Some of the cities studied are very wide, so a meteorological phenomenon may not affect different parts of the city equally. Since treating each subscriber individually can be unmanageable because of the amount of data that would have to be processed, a study could be done by zip code.

Both performance and meteorological information represent daily average values, so a temporary outage caused by a storm would be difficult to detect. Detailed information by hour might be more useful for the analysis.

It would also be useful to add call center information and technician dispatches to the customer's home. This, in addition to being an unequivocal indicator of poor quality in the network connection is a parameter that adds a crucial variable to the study, costs.

Since each call or dispatch has an associated cost with it, a new scenario is presented with many working points. Is it more expensive to fix a line in perfect conditions or not to act with a customer experiencing an unstable connection? Also, there is a very clear business case to present to an operator, how much does the weather increase costs?

1. Introducción

Hoy en día el mundo no puede entenderse sin las telecomunicaciones. La transmisión instantánea de información ha cambiado la forma de entender la sociedad. Las relaciones personales, los modelos de negocio o los hábitos de consumo son sólo unos ejemplos de acciones del día a día que se han visto afectadas por la llegada de Internet a la vida de las personas.

Este Trabajo de Fin de Grado (TFG) se centra en la predicción de deterioros en la calidad de conexión en las líneas de ADSL y VDSL teniendo en cuenta datos meteorológicos de las diferentes ciudades para las que se tiene información debido a que se sospecha que los distintos sucesos atmosféricos pueden tener un impacto negativo en la calidad de las comunicaciones.

El objetivo del presente proyecto tiene dos aplicaciones. En el aspecto corporativo, la idea es intentar satisfacer un caso de negocio para los distintos proveedores de red que consiste en la reducción de costes por llamadas de clientes al servicio técnico y despachos técnicos en la casa del cliente. En el aspecto personal, se busca adquirir conocimiento en todo el mundo de la ciencia de datos, incluyendo las diferentes librerías de analítica de datos para el lenguaje Python o el lenguaje SQL para obtener la información necesaria de la base de datos.

No hay que olvidar que una parte muy importante de cualquier proyecto de esta índole debe basarse en un conocimiento de la materia sobre la que se va a realizar el análisis. La experiencia cercana a dos años trabajando en una empresa del sector ha permitido al desarrollador de este proyecto conocer el detalle de los procedimientos de los diagnósticos que se realizan y la importancia a nivel comercial que puede tener cada una de las situaciones que pueden ocasionar un mal rendimiento de las líneas.

Realizar un proyecto de análisis de datos desde el principio al final servirá para coger soltura con las herramientas al alcance, a conocer todas las complicaciones que pueden surgir y a interiorizar buenas prácticas en el desarrollo.

1.1. Motivación

Como se comentaba antes, la conexión a Internet ha cambiado la sociedad en los últimos años. Esto ha originado múltiples posibilidades de negocio, especialmente para las empresas operadoras, y ha convertido en el sector en uno de los más rentables.

La motivación de este proyecto se puede dividir en tres aplicaciones:

- A nivel empresarial: Para las empresas proveedoras de servicios de comunicaciones es vital una buena calidad en el servicio prestado en sus clientes para mantener su modelo de negocio.

Los dos aspectos más importantes en términos económicos son la fuga de clientes y las intervenciones en la localización de la instalación. Si, tras el desarrollo del proyecto, se lograra predecir con más exactitud la inestabilidad en las líneas de clientes, aparecería la posibilidad de realizar intervenciones preventivas en el terreno.

De este modo, se resolverán incidencias antes de que el cliente detectara una degradación en el servicio reduciéndose el gasto derivado de llamadas a atención al cliente y evitando redundancia en intervenciones en caso de fallo masivo, a la vez que mejorando la imagen corporativa.

- A nivel personal: La elección de estudiar el Grado en Ingeniería en Tecnologías de Telecomunicación en 2012 ya traía consigo la motivación de formar parte de un sector como las telecomunicaciones, a la vanguardia tecnológica y con una fuerte presencia en el día a día de la sociedad.

  A este interés se le suma la atracción que supone un sector con importancia tan creciente como la ciencia de datos. Con este trabajo se pretende realizar un proyecto de analítica de datos desde el principio hasta el final y, con ello, la familiarización con las técnicas y tecnologías que se deben utilizar, así como los pasos a seguir y las buenas prácticas que se deben tomar.

- A nivel social: Dada la importancia que están tomando las actividades que requieren de conexión a Internet a nivel social, cultural, educativo o burocrático, entre otros ejemplos, se pueden dar situaciones de desigualdad social cuando el no poder costear una conexión a Internet de calidad afecte negativamente en la igualdad de oportunidades.

  Por ello, la mejora en las prestaciones a nivel de red busca también dotar de recursos digitales a la sociedad que puedan ser utilizados como herramientas de cohesión social. Un ejemplo de estas pueden ser la educación o asistencia sanitaria a distancia en zonas rurales.

1.2. Objetivos

El negocio de las empresas operadoras de red se basa en la venta de servicios de telefonía e Internet tanto a empresas como a particulares.

Los ingresos que perciben estas compañías tienen relación directa con el número de clientes y el precio de los productos que vende. Los gastos son más diversos, pueden ir desde el establecimiento de nuevas redes, la renovación y solución de problemas en redes ya implementada hasta la atención al cliente.

El caso de negocio que se persigue en este proyecto tiene por objetivo el maximizar los ingresos y reducir los costes. Para ello, el problema principal a solventar es la mala calidad en la conexión que pueden percibir los clientes.

Un mal rendimiento de las líneas de datos supone, por una parte, una reducción de los ingresos de una compañía al aumentar el riesgo de fuga de los clientes y un

empeoramiento en la imagen corporativa al no satisfacer las necesidades por las que se contratan los servicios.

Por otra parte, se aumentan los costes derivados de la atención al cliente, especialmente el aumento de llamadas al servicio de atención al cliente y un mayor número de incidencias a resolver en el lugar donde se producen.

Este proyecto en concreto se centra en estudiar la relación que puede existir entre los sucesos meteorológicos que ocurren en una ciudad y la calidad de las líneas en esa ciudad.

El modelo a desarrollar durante esta investigación permitiría conocer de antemano las probabilidades de aumento de incidencias en una zona concreta partiendo de la información meteorológica.

Con ello, se podría llevar a cabo una política proactiva de resolución de incidencias, antes de que el cliente perciba inestabilidad en la conexión, en vez de reactiva, tomando acciones cuando el cliente informa de la situación.

1.3. Contenido de la memoria

La presente memoria está organizada de forma que sea lo más comprensible posible. Por otro lado, se pretende seguir la estructura que debe tener un proyecto de aprendizaje máquina, explicado posteriormente en el apartado 2.6.

El apartado 2 responde al primer paso del proceso que es el conocimiento de la tecnología con la que se quiere trabajar.

Los apartados 3 y 4 se corresponden con el conocimiento de los datos que se disponen, tanto de rendimiento de líneas como meteorológicos.

En el apartado 5 se realiza el procesado de los datos para convertirlos en adecuados para la creación de los modelos que aparece en el apartado 6.

En el capítulo 8 se realiza la valoración de los resultados obtenidos y en el capítulo 9 se definen los puntos futuros a seguir para mejorar el análisis.

2. Estado del arte

En este capítulo se expone el estado del arte en el que se ha llevado a cabo este proyecto. La finalidad es investigar y presentar las tecnologías y técnicas más utilizadas en el campo en el que se desarrolla el proyecto.

2.1. Redes ADSL

La tecnología DSL ha supuesto el servicio de Internet de banda ancha más utilizado en los últimos años. El ADSL (Asymetric Digital Suscriber Line) [2] ha sido la opción más común especialmente para uso particular debido a que sus características se amoldan especialmente al uso de Internet que se hace en los hogares. Permite una velocidad de hasta 20 MBits/s utilizando la línea telefónica tradicional, lo que obliga a la colocación de microfiltros en los dispositivos telefónicos para evitar la interferencia de las señales de voz y datos. La velocidad máxima de bajada es inversamente proporcional a la longitud de bucle, que es la longitud del cable que conecta la central con el domicilio del cliente. La velocidad de subida puede ser hasta diez veces menor, debido a la principal característica de esta tecnología que es la asimetría en el ancho de banda de subida y bajada.

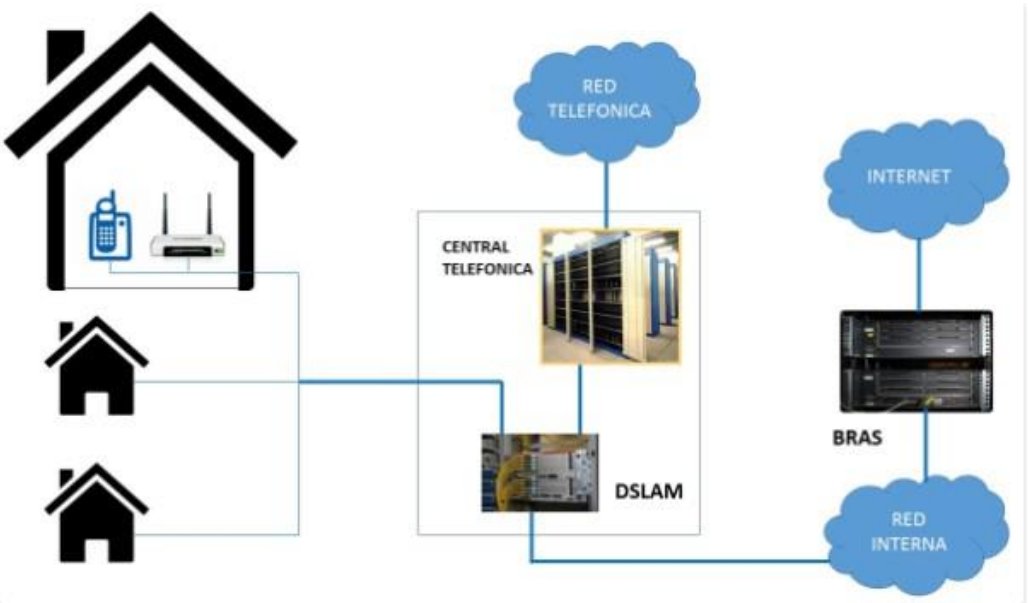

Figura 2-1: Esquema de una red ADSL [2]

2.2. BRAS

Siglas de «Broadband Remote Access Server» [2] y representa un nodo intermedio entre una red ADSL e Internet al estar conectado a los DSLAM de la red. El BRAS asigna la dirección IP a cada usuario dentro de la red que será utilizada para la conexión a Internet. El BRAS dispone de un rango de direcciones IP que puede asignar a los clientes, que siempre es menor al número de estos. Esto es

posible gracias a la asignación dinámica de direcciones IP. El tratamiento del rango de direcciones, que cada operador tiene asignado en exclusiva, es una solución eficaz a un problema como la escasez de direcciones IP debido al creciente desarrollo del sector de las telecomunicaciones.

2.3. DSLAM

Son las siglas de «Digital Subscriber Line Access Multiplexer» [2]. Es el equipo que interconecta el BRAS con las líneas de cada cliente. A este se conectan las líneas telefónicas de los abonados que son tratadas de manera independiente en función del servicio contratado, por ejemplo, limitando la velocidad de transmisión de datos.

Debido a que la distancia entre el DSLAM y el CPE tiene un impacto negativo en la máxima velocidad alcanzable en el domicilio de un abonado, es importante para la empresa operadora que la situación geográfica de cada DSLAM se decida con el objetivo de optimizar la distancia a un mayor número de clientes.

2.4. CPE

Las siglas de «Customer Premise Equipment» [2] describen el router instalado en el domicilio del cliente y es el nodo final de la red ADSL para pasar a considerarse red privada. Haciendo uso del mecanismo NAT el CPE asigna una dirección IP privada a cada terminal dentro de la subred utilizando una única IP pública para la salida a Internet.

2.5. Ciencia de datos

Este trabajo está pensado para ser un proyecto de ciencia de datos (Data Science).

Según Drew Conway [3] la ciencia de datos es la conjunción de varias áreas de conocimiento, como se muestra en el siguiente gráfico.

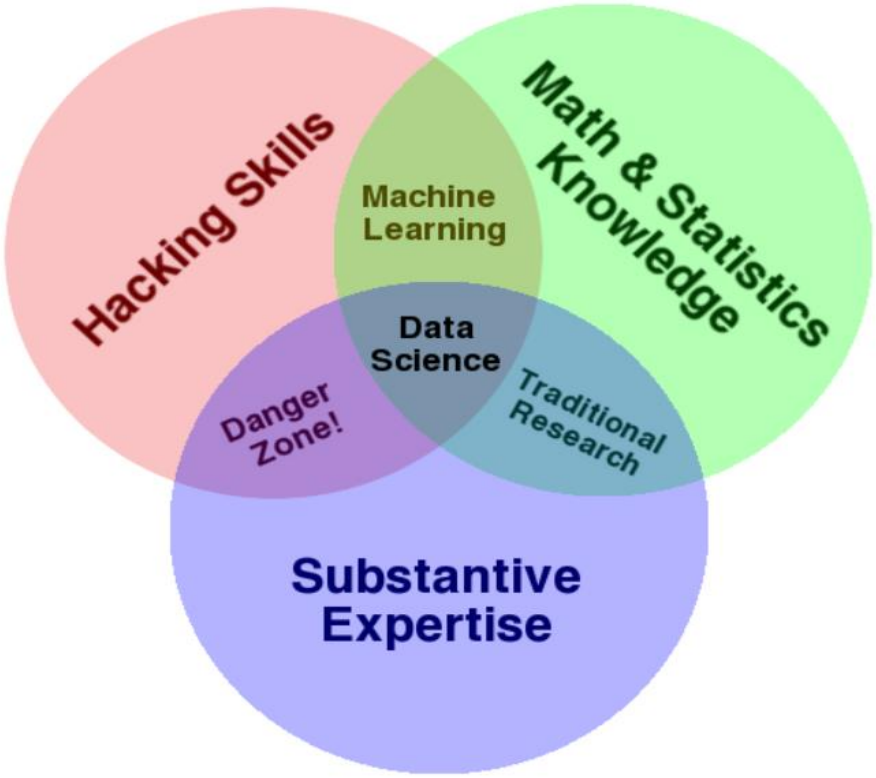

Figura 2-2: Diagrama de Venn de Drew Conway sobre ciencia de datos

A continuación, se detallan los campos del conocimiento presentes el diagrama:

- Habilidades informáticas: Necesarias para trabajar con grandes volúmenes de información que debe ser adquirida, limpiada y manipulada.
- Conocimiento matemático y estadístico: Permite a un científico de datos elegir los métodos y herramientas adecuadas para extraer información de los datos.
- Conocimiento sobre la materia: En un campo científico es necesario para hacer las preguntas correctas y concisas e interpretar las respuestas obtenidas.

Con la combinación de dos de ellos se llega a situaciones diferentes:

- La conjunción de conocimientos matemáticos y estadísticos y un conocimiento experto en la materia de estudio es lo denominado como investigación tradicional, sin el uso de herramientas de tratamiento masivo de datos.
- El aprendizaje máquina requiere de habilidades informáticas y un conocimiento de métodos matemáticos y estadísticos, pero no de motivación científica.
- Lo denominado por Drew Conway como zona de peligro es la combinación de poseer conocimiento en la materia de estudio y la capacidad de tratamiento de los datos sin el conocimiento matemático.

La aplicación de métodos no adecuados puede concurrir en análisis erróneos.

La suma de las tres habilidades descritas arriba es la ciencia de datos.

2.6. Aprendizaje máquina o computacional

El aprendizaje máquina es la ciencia que enseña a las máquinas a pensar como un humano y perfeccionar su conocimiento de manera autónoma a través de información del mundo real.

CRISP-DM (Cross Industry Standard Process for Data Mining) [4] es el proceso más aceptado y utilizado para la realización correcta de un proyecto de aprendizaje máquina. En el siguiente diagrama se muestran todas las fases del proyecto y las relaciones entre ellas:

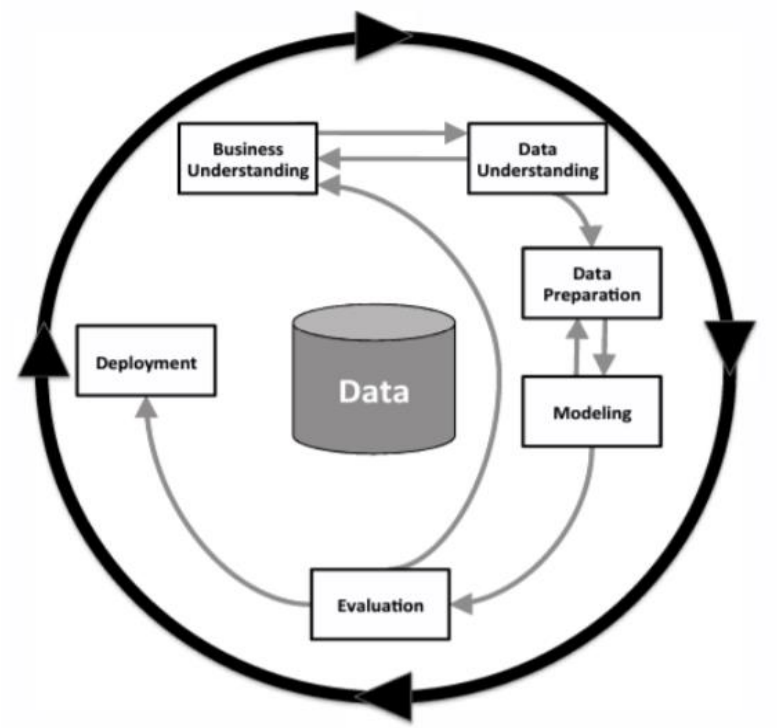

Figura 2-3: Diagrama del proceso CRISP-DM [4]

- El conocimiento del negocio: Los proyectos de análisis de datos se centran en casos de negocio como la obtención de nuevos clientes, reducción de costes u optimización de procesos de fabricación de productos y no en la construcción en sí de un modelo de predicción. Por lo tanto, una primera fase de cualquier proyecto de análisis consistirá en

comprender plenamente el problema de negocio que se está abordando y, posteriormente, diseñar una solución de análisis de datos para él.

- Comprensión de los datos: Una vez definida la manera en la que se utilizará el análisis predictivo de datos para abordar el problema en cuestión, es importante que el analista comprenda las diferentes fuentes de datos disponibles y los tipos de datos que contienen estas fuentes.
- Preparación de datos: Cada modelo de analítica de datos requiere de tipos de datos específicos. Esta fase de CRISP-DM incluye todas las actividades requeridas construir un conjunto de datos idóneo para en el análisis combinando la información de las diferentes fuentes.
- Modelado: Durante la fase de modelado se utilizan diferentes algoritmos de aprendizaje máquina para crear una variedad de modelos de predicción, de los cuales se tomará el mejor para la implementación
- Evaluación: Antes de que los modelos sean desplegados para la utilización por parte del usuario final, es importante que sean evaluados y demostrar si son adecuados para el propósito definido en la primera fase.
- Despliegue: Ya que los modelos creados tienen la finalidad de satisfacer un caso de uso, la última fase de CRIPS-DM comprende todo el trabajo de integración con éxito del modelo para ser utilizado por el usuario final.

La figura 2.3 también muestra el flujo entre cada una de las fases y destaca que los datos están en el centro del proceso. Algunas fases del proceso están más vinculadas entre sí que otras. La conexión bidireccional del conocimiento del negocio y la comprensión de los datos es una muestra de ello, ya que los proyectos pueden pasar tiempo trasladándose entre ambas fases; un caso de uso debe disponer de datos suficientes para permitir un modelado adecuado y el descubrimiento o procesado de nuevos datos puede crear nuevas oportunidades a abordar.

Por otro lado, las fases de preparación y modelado de datos también están estrechamente relacionadas, ya que cada modelo puede necesitar de un procesado distinto de datos. Una vez en el proceso de evaluación se debe volver a la fase de comprensión del negocio para determinar si la salida del proceso satisface las necesidades marcadas en la definición del proyecto.

Por último, tras el despliegue, se especifica con la flecha gruesa que se encuentra en el exterior de la figura que el modelado es un proceso continuo, siempre está en proceso de ajuste y mejora.

2.7. Modelado de datos

El objetivo de un modelo es proporcionar un resumen simple de un conjunto de datos. Idealmente, el modelo descartará el ruido y logrará identificar la información relevante [5].

Pese poder parecer un análisis algo simple, los modelos se pueden dividir en dos grandes grupos: Modelos de aprendizaje supervisado o no supervisado [6].

La principal diferencia entre ambos radica en los datos de entrada.

- El aprendizaje supervisado utiliza un etiquetado inicial de cómo es la salida en función de los datos de entrada (por ejemplo, un día con una determinada temperatura, presión atmosférica y fuerza del viento es un día soleado-, por tanto, el objetivo es la creación de una función que, dado unos datos de entrada, prediga el valor de salida con la mayor aproximación posible.

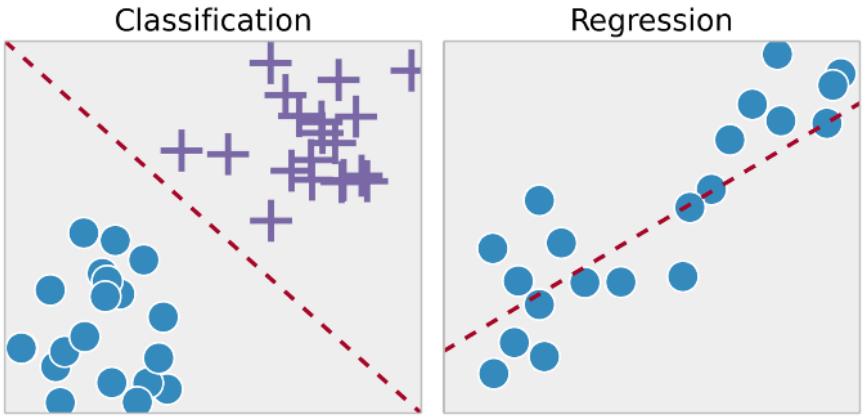

Figura 2-4: Tipos de aprendizaje supervisado [6]

El aprendizaje supervisado normalmente se utiliza en el contexto de la clasificación, cuando se pretende asignar una etiqueta de salida en función de unos valores de entrada, o regresión, cuando se asigna un valor continuo de salida a los valores de entrada.

En ambos contextos lo que se pretende es encontrar relaciones específicas en los datos de entrada que permitan generar el correcto valor de la salida de manera efectiva.

Los algoritmos más comunes de este tipo de aprendizaje incluyen regresión logística, máquinas vector soporte (SVM), redes neuronales o árboles de decisión.

- El aprendizaje no supervisado tiene como objetivo conocer la estructura natural del conjunto de datos sin utilizar etiquetas explícitas de ello [6].

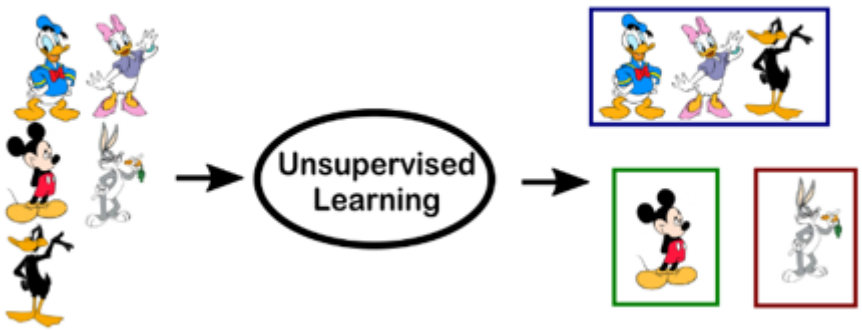

Figura 2-5: Ejemplo de funcionamiento de algoritmos de clustering [6]

Dos casos de uso comunes del aprendizaje no supervisado son el análisis exploratorio y la reducción dimensional.

- A la hora de realizar un análisis exploratorio es posible apoyarse en el aprendizaje no supervisado para identificar automáticamente la estructura de datos. Un ejemplo de ello es el agrupamiento (en inglés clustering) que permite la segmentación de los datos, a menudo inabarcables para un humano, como un paso inicial para la extracción de información de un conjunto de datos [6].
- La reducción de dimensionalidad consiste en la representación de un conjunto de datos con un número menor de variables analizando la relación entre las variables de entrada, eliminando información con poco interés o redundante [6].

Los algoritmos más comunes de aprendizaje no supervisado incluyen agrupamiento k-medias o análisis de componente principal (PCA) [6].

2.8. Validación del modelo

Tradicionalmente, el enfoque del modelado se centra en la inferencia, generación de hipótesis y demostración de si es correcta [5].

Según Grolemund y Wickham en [5] hay unos preceptos que respetar para realizar una inferencia correctamente:

- Cada observación debe utilizarse para exploración (o entrenamiento) o para confirmación, pero nunca ambas.
- Una observación se puede utilizar las veces necesarias para la exploración, pero una única vez para verificación.

Si se pretende tener esto en cuenta se recomienda dividir el conjunto de datos en varios grupos que tendrán un tratamiento distinto [5]:

- Datos de entrenamiento (60%): Con estos datos se podrán realizar los ajustar los modelos que se quieran aplicar.
- Datos de validación (20%): Con estos datos se validará la eficacia de cada uno de los modelos aplicados, pudiéndose realizar modificaciones o ajustes para maximizar la tasa de acierto de los algoritmos.
- Datos de verificación (20%): Estos datos se utilizarán una única vez para probar la eficacia del modelo final.

2.9. Marco regulador

En este apartado se describe el marco regulador en el que se ha desarrollado este proyecto.

Principalmente el proyecto está basado en dos regulaciones: El Reglamento General de Protección de Datos (RGPD) que recoge la protección de las personas

físicas en cuanto al tratamiento y libre circulación de los datos personales [7] y la LOPDGDD que tiene por objetivo adaptar el reglamento europeo en esta materia a la legislación española [8] .

La RGPD supone una actualización de la anterior ley europea de protección de datos de carácter personal LOPD que creó un marco regulador que buscaba establecer un equilibrio entre la protección de la vida privada de las personas y la libre circulación de datos personales dentro de la Unión Europea.

Entre las principales novedades que establece la RGPD es la obligación de que todas las empresas y organizaciones que traten datos de ciudadanos europeos se deben regir por la misma legislación en toda la UE, evitando diferencias de criterios según el país en el que se realice el tratamiento.

Otro punto relevante que se introduce es la necesidad de que exista un consentimiento libre e informado de la persona cuyos datos son utilizados. Las organizaciones deben analizar los datos que se tratan, la finalidad con la que lo hacen y las operaciones que llevan a cabo. Más adelante deben poder demostrar que el interesado ha comprendido el consentimiento concedido [9].

Aparte de estos puntos se establece también el derecho al acceso a una copia de los datos que están siendo tratados, el derecho al olvido o la solicitud del cese de operaciones del tratamiento de sus datos.

La LOPGDD [10] adapta el Derecho interno español al Reglamento General de Protección de Datos. Además, trata los puntos que debe ser regulados por cada Estado.

Un ejemplo de regulación competente de cada Estado es el acceso a los datos personales de personas fallecidas por parte de personas vinculadas u otras personas o instituciones a las que se haya designado expresamente para ello. Por otro lado, se establece la edad mínima para el consentimiento de tratamiento de datos a los 14 años y se crea una categoría especial de datos [11] para los correspondientes a la ideología, origen étnico, orientación sexual, religión o cualquier información que pudiera ocasionar cualquier tipo de discriminación.

Cabe comentar que los datos tratados a lo largo de este proyecto correspondientes a la información de las líneas telefónicas y de Internet se encuentran anonimizados. De esta forma no es posible obtener información personal de un abonado a partir del servicio de telecomunicaciones que contrata.

3. Información del escenario

Los datos de rendimiento de las líneas están presentes en la tabla ***"PERFORMANCE"***. Entre ellos aparecen datos de distinto tipo y categoría que incluyen información del abonado como el identificador, el DSLAM al que está conectado, el servicio contratado; también aparecen datos operacionales como la potencia de transmisión y recepción de señal y una serie de alarmas de evaluación de la calidad de la línea como la detección de fallo en el enlace o error en el impulso.

Esta información se obtiene diariamente obteniendo datos operacionales de las líneas de cada abonado y realizando diagnósticos a partir de ella.

CAMPO	**TIPO**	**DESCRIPCION**
INSERTION_DATE	DATE	Fecha a la que pertenecen los datos recolectados.
LINE_ID	VARCHAR2(128)	Número de identificación del abonado. (Anonimizado).
DSLAM	VARCHAR2(128)	DSLAM al que está conectada la línea del cliente.
SERVICE_PRODUCT	VARCHAR2(128)	Servicio comercial que se tiene contratado. En función del cliente la nomenclatura varía, pero se suele indicar la velocidad de bajada contratada y algún elemento identificador definido por el cliente.
DSLAM_TYPE	VARCHAR2(30)	Modelo del DSLAM.
CARD_TYPE	VARCHAR2(128)	Tipo de tarjeta a la que está conectado el cliente.
CARD_VERSION	VARCHAR2(90)	Versión de tarjeta.
SYSTEMTYPE	NUMBER(10,0)	Estándar de sincronismo.
CURRENTRATEDS	NUMBER(10,0)	Velocidad de bajada que se detecta en la línea.
CURRENTRATEUS	NUMBER(10,0)	Velocidad de subida que se detecta en la línea.
POWERDS	FLOAT(126)	Potencia en bajada detectada.
POWERUS	FLOAT(126)	Potencia en subida detectada.
ATTENUATIONDS	FLOAT(126)	Atenuación de la señal en bajada que sufre la línea.
ATTENUATIONUS	FLOAT(126)	Atenuación de la señal en subida que sufre la línea.
CPE_CHIPSET	VARCHAR2(80)	Chipset del CPE del abonado.
CPE_VERSION	VARCHAR2(80)	Versión del software del CPE.
CPE_MODEL	VARCHAR2(80)	Modelo del CPE del abonado.
INTRATECH_RECOMMENDATION	VARCHAR2(80)	Recomendación de nuevo servicio dentro de la misma tecnología.
INTERTECH_RECOMMENDATION	VARCHAR2(80)	Recomendación de nuevo servicio incluyendo otra tecnología.
MABR_DS	NUMBER(10,0)	Máxima velocidad de bajada alcanzable por la línea.

STABILITY	NUMBER(3,0)	Indicador de estabilidad en la conexión [0-3]. El valor 9 indica que no hay valores suficientes para determinar la estabilidad.
BADSPLICE_DET	NUMBER(1,0)	Alarma de fallo en el empalme.
BRIDGED_TAP_DET	NUMBER(10,0)	Alarma de conexión puenteada.
UNBALANCED_DET	NUMBER(1,0)	Alarma de cableado desbalanceado.
HPN_DET	NUMBER(1,0)	Alarma de ruido de alta potencia.
HPNHF_DET	NUMBER(1,0)	Alarma de ruido de alta potencia y frecuencia.
IMPULSE_DET	NUMBER(1,0)	Alarma de ruido impulsivo.
MISSING_MICROFILTER_DET	NUMBER(1,0)	Alarma de falta de microfiltro.
LOOP_LENGTH	NUMBER	Longitud de bucle entre el abonado y el DSLAM.
LS_DISPATCH_SCORE	NUMBER(3,0)	Código de recomendación de resolución de avería por medio de un técnico.
DETAIL_ACTION_1	VARCHAR2(128)	Descripción del código de recomendación de resolución de avería.

Tabla 3-1: Descripción de los campos de rendimiento de líneas.

4. Datos meteorológicos obtenidos

Los datos obtenidos son del National Climatic Data Center (NCDC) [12], institución pública que dispone de una gran cantidad de datos climáticos para estaciones de todo el mundo. Se han buscado datos para las ciudades de las que se disponen datos de rendimiento para las líneas.

Su descripción [13] es la siguiente

Campo	Posición	Tipo	Descripción
STN---	1-6	Entero	Número de estación para su localización
WBAN	8-12	Entero	Número de oficina meteorológica de la fuerza aérea marina.
YEAR	15-18	Entero	Año
MODA	19-22	Entero	Mes y día
TEMP	25-30	Real	Temperatura media en el día en grados Fahrenheit. En caso de no aparecer = 9999.9
Count	32-33	Entero	Número de observaciones utilizadas para calcular la temperatura media
DEWP	36-41	Real	Punto de rocío medio en el día en grados Fahrenheit. En caso de no aparecer = 9999.9
Count	43-44	Entero	Número de observaciones utilizadas para calcular el punto de rocío.
SLP	47-52	Real	Presión media a nivel de mar en el día en milibares. En caso de no aparecer = 9999.9
Count	54-55	Entero	Número de observaciones utilizadas para calcular la presión a nivel del mar.
STP	58-63	Real	Presión media en la estación en el día en milibares. En caso de no aparecer = 9999.9
Count	65-66	Entero	Número de observaciones utilizadas para calcular la presión en la estación.
VISIB	69-73	Real	Visibilidad media en el día en millas. En caso de no aparecer = 999.9
Count	75-76	Entero	Número de observaciones utilizadas para calcular la visibilidad media.
WDSP	79-83	Real	Velocidad media del viento en el día en nudos. En caso de no aparecer = 999.9
Count	85-86	Entero	Número de observaciones utilizadas para calcular la velocidad media del viento.
MXSPD	89-93	Real	Máxima velocidad sostenida del viento en el día en nudos. En caso de no aparecer = 9999.9
GUST	96-100	Real	Ráfaga de viento máximo detectada en nudos.
MAX	103-108	Real	Temperatura máxima detectada en el día en grados Fahrenheit. En caso de no aparecer = 9999.9

Flag	109-109	Carácter	Espacio en blanco indica que la temperatura máxima fue obtenida del reporte de temperatura máxima. Asterisco(*) indica que la temperatura máxima fue derivada de los datos por hora.
MIN	111-116	Real	Temperatura mínima detectada en el día en grados Fahrenheit. En caso de no aparecer = 9999.9
Flag	117-117	Caracter	Espacio en blanco indica que la temperatura mínima fue obtenida del reporte de temperatura mínima. Asterisco(*) indica que la temperatura mínima es derivada de los datos por hora.
PRCP	119-123	Real	Precipitación total (lluvia y/o aguanieve) detectada durante el día en pulgadas y centésimas. Puede no devolver 0 en caso de no '99.99'.
Flag	124-124	Caracter	A = 1 informe de precipitación de 6 horas. B = Suma de 2 informes de 6 horas. C = Suma de 3 informes de 6 horas. D = Suma de 4 informes de 6 horas. E = 1 informe de precipitación de 12 horas. F = Suma de 2 informes de 12 horas. G = 1 informe de precipitación de 24 horas.
SNDP	126-130	Real	Profundidad de nieve en pulgadas.
FRSHTT	133-138	Entero	Indicadores (6 dígitos): Niebla: Primer dígito Lluvia: Segundo dígito Nieve: Tercer dígito Granizo: Cuarto dígito Truenos: Quinto dígito Tornado: Sexto dígito

Tabla 4-1: Descripción de los campos de información meteorológica

5. Pre-procesado de datos

5.1. Datos climatológicos

En primer lugar, se importan los datos unificados del clima.

```
weather = pd.read_csv('Clima/clima.txt')
```

Se ajustan los nombres de las columnas para ser más legible.

```
weather.columns = weather.columns.str.strip()
weather = weather.rename(index=str, columns={"STN---": "CODE"})
weather['CODE'] = weather['CODE'].astype(str)
```

Los campos MAX y MIN pueden contener el caracter asterisco como se indica en la tabla de especificación de los campos. Estos deben ser eliminados para tratar el campo con formato de coma flotante en lugar de cadena de caracteres:

```
     YEARMODA     MAX    MIN
0    20180101.0   73.4*  64.4*
1    20180102.0   78.8*  63.7
2    20180103.0   82.6*  64.4
3    20180104.0   82.4*  63.9*
4    20180105.0   80.6*  63.7
5    20180106.0   80.6*  65.7*
6    20180107.0   80.6*  64.4*
7    20180108.0   82.4*  62.6
8    20180109.0   73.4*  64.4
9    20180110.0   80.6*  60.8*
```

```
weather['maximum'] = weather['MAX'].astype(str).str[:4].astype(float)
weather['minimum'] = weather['MIN'].astype(str).str[:4].astype(float)
```

El campo FRSHTT necesita dos modificaciones: sustituir todos los valores nulos por 0 y asegurarse de que los datos son tratados como cadena de caracteres y no como número:

```
weather['FRSHTT'] = weather['FRSHTT'].fillna(0)
weather['FRSHTT'] = weather['FRSHTT'].astype(int).astype(str)
weather['FRSHTT'] = weather['FRSHTT'].str.zfill(6)
```

Para el resto de variables se modifican los valores indicados por el proveedor de datos como datos faltantes para pasar a ser considerados como nulos

```
weather = weather.replace(9999.9, np.NaN)
weather = weather.replace(999.9, np.NaN)
weather = weather.replace("99.99 ", np.NaN)
```

En la columna PRCP, referente a la precipitación, también pueden aparecer valores alfanuméricos que deben ser eliminados.

```
weather.PRCP = weather.PRCP.str.replace(r"[a-zA-
Z]",'').astype('float')
```

Luego se eliminan las columnas inservibles:

```
weather = weather.drop(['MAX','MIN','', '.1','.2','.3',
'.4','.5','Unnamed: 22'],axis=1)
```

Se añade la ciudad a la que corresponden los datos. Para ello se crea un archivo ***stations.txt*** en el que aparece la información de a qué ciudad corresponde cada código de estación, esta es cruzada con la información del clima.

```
CITY,CODE
CITY_A,802110
CITY_B,800280
CITY_C,802220
CITY_D,800220
CITY_E,802140
CITY_F,803150
CITY_G,803420
CITY_H,800970
CITY_I,800090
CITY_J,802590
```

```
stations = pd.read_csv('stations.txt')
stations['CODE'] = stations['CODE'].astype(str)
weather = pd.merge(weather, stations, on = 'CODE')
weather['CITY'] = weather['CITY'].astype(str)
```

Por confidencialidad, los datos de cada ciudad son anonimizados.

También se crea la columna “fecha” que posibilita la visualización, ya que para la creación de algunos gráficos es necesario que la fecha se encuentre en formato datetime (YYYY-MM_DD).

```
weather['fecha'] =
pd.to_datetime(weather['YEARMODA'].astype(str).map(lambda x: x[:4]
+'-'+ x[4:6]+'-'+ x[6:8]))
```

Una vez realizadas las modificaciones necesarias se procede a realizar visualizaciones exploratorias de los datos climatológicos:

- Temperatura por ciudad:

```
sns.lineplot(x="fecha", y="TEMP", hue="CITY",data= weather)
```

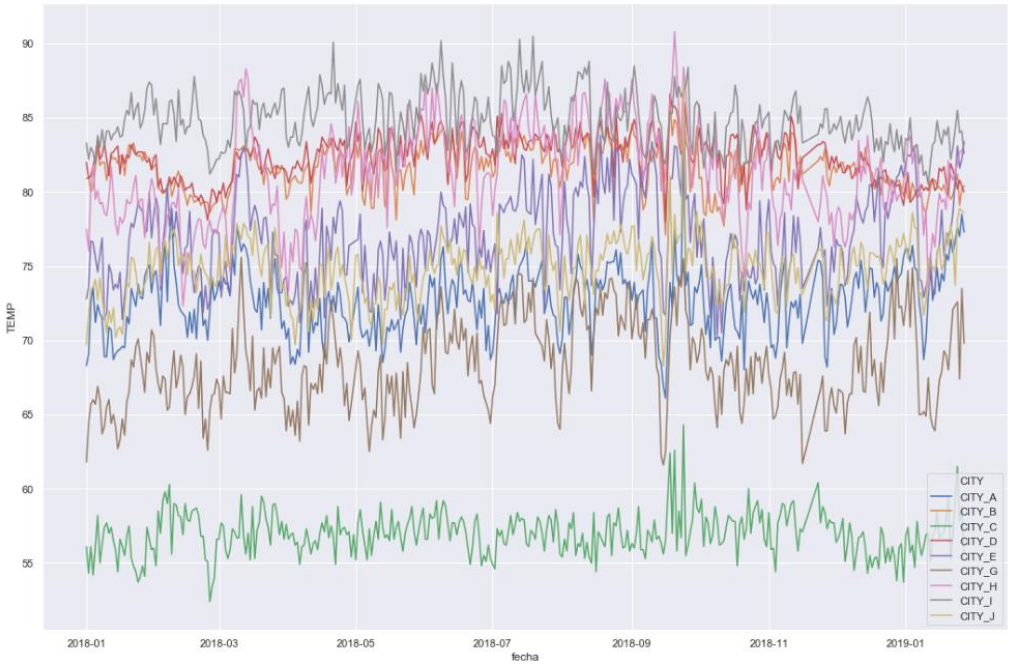

Figura 5-1: Evolución de temperatura por ciudad

Se aprecia como la mayoría de las ciudades siguen un patrón similar. Destaca la ciudad ***CITY_B*** con unos valores de temperatura más bajos de lo habitual.

- Sucesos climatológicos:

```
sns.countplot(x='FRSHTT', data= weather)
```

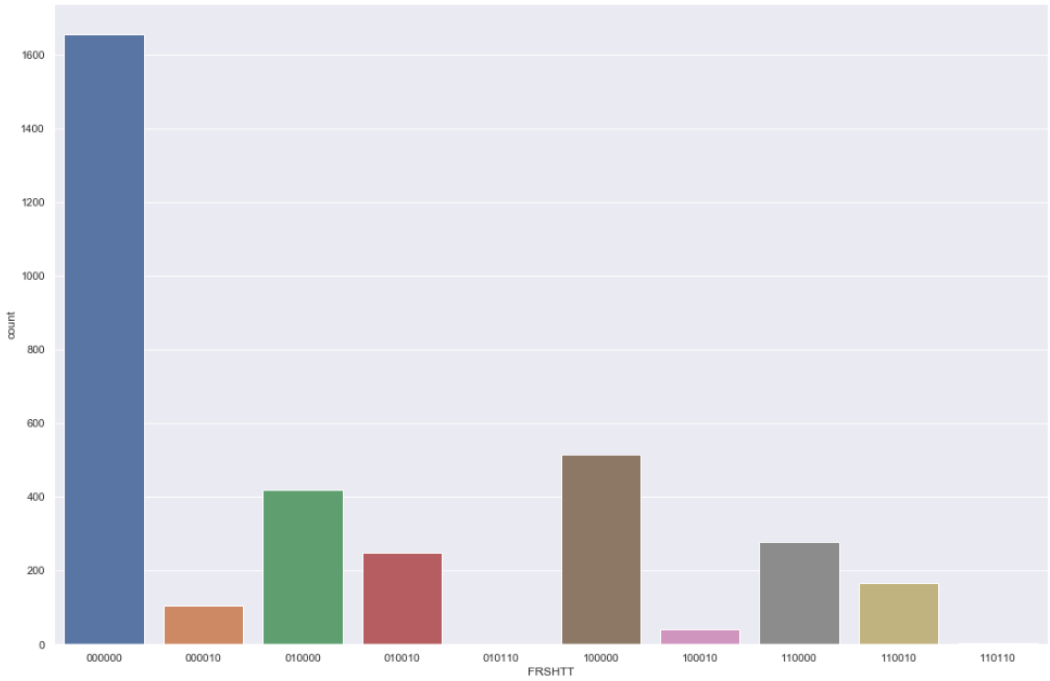

Figura 5-2: Histograma de sucesos climatológicos

Se ve como el panorama más habitual es soleado, el segundo es nublado y el tercero lluvioso.

Por simplicidad, se unifican estos indicadores en tres escenarios: Sol, lluvia y tormenta.

```
d = {'FRSHTT': ['000000','000010','010000','010010','010110',
'100000','100010','110000','110010','110110'], 'Cat': ['Sol',
'Tormenta','Lluvia','Tormenta', 'Tormenta','Lluvia','Tormenta',
'Lluvia','Tormenta','Tormenta']}

df = pd.DataFrame(data=d)

weather = pd.merge(weather, df, on = 'FRSHTT')
weather['cat'] = weather['Cat'].astype(str)
```

Luego se muestra cómo se distribuyen las diferentes situaciones en función de la ciudad.

```
sns.countplot(x='CITY', hue = 'cat', data= weather)
```

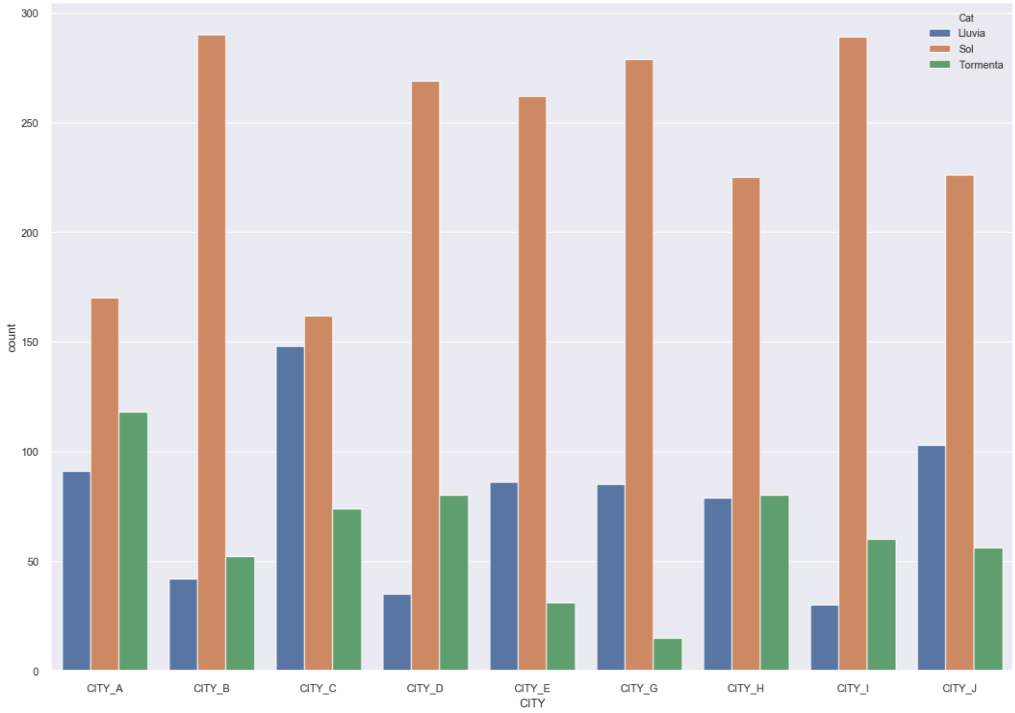

Figura 5-3: Situaciones climatológicas por ciudad

Se muestra cómo las ciudades más soleadas son CITY_B y CITY_I, la ciudad con mayor porcentaje de días lluviosos es CITY_C y la ciudad con mayor probabilidad de sufrir una tormenta es CITY_A.

5.2. Datos de rendimiento de líneas

Los datos de rendimiento son generados cada día conteniendo toda la información descrita en la tabla 3-1 para cada línea ADSL o VDSL de cada cliente.

Para el caso del cliente del estudio se trata de 1 millón de líneas diarias, aproximadamente 250 MB por archivo.

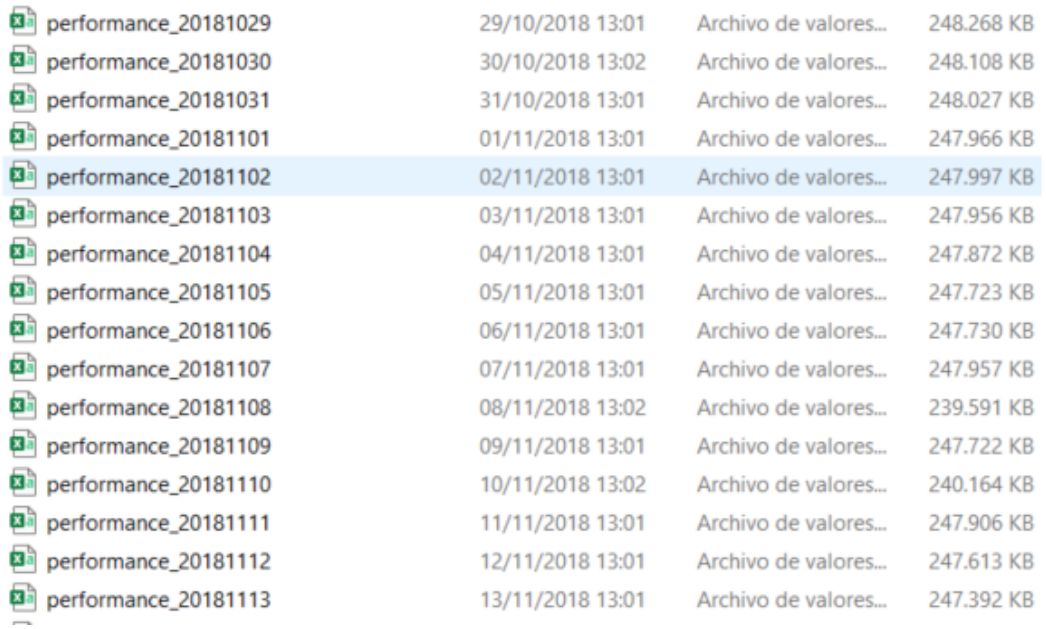

Ilustración 5-1: Muestra de ficheros de rendimiento de líneas

Como el volumen de datos es tan grande y el objetivo del análisis se centra en cómo afecta la climatología a la estabilidad de las líneas a nivel de ciudades, se ha implementado un script ***summarize.py*** para agrupar los datos por ciudad y fecha.

En él se añadirá la información de ciudad a los datos de rendimiento. Esta información se obtiene de la tabla ***NEIGHBORHOOD_INFO*** donde, mediante el proceso de provisión, el cliente indica la localización de cada línea.

```
SELECT * FROM (
SELECT NBHOOD_CITY,dslam, count(*) LINEAS FROM V_NEIGHBORHOOD JOIN
v_ports USING (line_id)
WHERE NBHOOD_CITY IN (CITY_A,'CITY_B,'CITY_C,'CITY_D',
'CITY_E','CITY_F', 'CITY_G','CITY_H','CITY_I', 'CITY_J')
GROUP BY NBHOOD_CITY,dslam)
WHERE LINEAS >10 ORDER BY 1, 3 desc;
```

El resultado se almacena en el fichero ***ciudad_dslam.csv***

```
$ head -10 ciudad_dslam.csv
"NBHOOD_CITY","DSLAM","LINEAS"
CITY_A,Galan_MA5600,807
CITY_A,Galan_4,699
```

```
CITY_A,CTRO-2,658
CITY_A,BOSQUE,650
CITY_A,ELCARMEN-2,632
CITY_A,ELCARMEN,629
CITY_A,Sur,599
CITY_A,DM_QUIALARSE_I_ISA02FD,586
CITY_A,SUR2,577
```

Tras las operaciones de limpieza de los datos y la adición de la ciudad para el agregado posterior se crean las variables ***YEARMODA*** y ***YEARMODA1*** que consisten en la fecha con el mismo formato que aparece en el fichero de datos climatológicos. El campo ***YEARMODA1*** no es más que la fecha del día siguiente.

Es posible que la climatología pueda influir en las líneas en el día en que ocurren los eventos, pero también que afecte a los resultados del día siguiente ya que la mayoría de los diagnósticos se llevan a cabo por la noche. Esto será un punto del análisis posterior.

Finalmente, este el código del script ***summarize.py***:

```
import glob
import pandas as pd
from  datetime import datetime
from datetime import timedelta

open('performance_data.csv','w+')
files = glob.glob("Performance/performance*")

city_dslam = pd.read_csv('ciudad_dslam.csv')

for file in files:
       lines = pd.read_csv(file)

       lines.columns =
       ['PAIS','INSERTION_DATE','line_id','DSLAM','service_product',
       'DSLAM_TYPE','card_type','card_version','SYSTEMTYPE',
       'CURRENTRATEDS','CURRENTRATEUS','POWERDS','POWERUS',
       'ATTENUATIONDS','ATTENUATIONUS', 'CPE_CHIPSET ','CPE_VERSION',
       'CPE_MODEL','INTRATECH_RECOMMENDATION','INTERTECH_RECOMMENDATION
       ','MABR_DS','stability',' BADSPLICE_DET','BRIDGED_TAP_DET',
       'UNBALANCED_DET','HPN_DET','HPNHF_DET','IMPULSE_DET',
       'MISSING_MICROFILTER_DET','Loop_Length','LS_DISPATCH_SCORE',
       'DETAIL_ACTION_1','FFTX_LINE']

       lines['DSLAM'] = lines['DSLAM'].astype(str)
       city_dslam['DSLAM'] = city_dslam['DSLAM'].astype(str)
       joined = pd.merge(lines, city_dslam, on='DSLAM')
       joined.columns = joined.columns.str.strip()
       joined = joined.rename(index=str, columns={"NBHOOD_CITY":
       "CITY"})
       joined['CITY'] = joined['CITY'].astype(str)
```

```
joined['YEARMODA']=pd.to_datetime(joined['INSERTION_DATE']).map(
lambda x: 10000*x.year +100*x.month+x.day)

joined['FECHAMAS1'] = pd.to_datetime(joined['INSERTION_DATE'])-
timedelta(days=1)

joined['YEARMODA1']=pd.to_datetime(joined['FECHAMAS1']).map(lamb
da x: 10000*x.year +100*x.month+x.day)

joined.filter(['YEARMODA','YEARMODA1']).drop_duplicates()
nuevo_df = joined.groupby(['CITY','INSERTION_DATE' ,'YEARMODA',
'YEARMODA1','stability']).agg({'line_id':'count','Loop_Length':'
mean','CURRENTRATEDS':'mean','CURRENTRATEUS':'mean','POWERDS':'m
ean','POWERUS':'mean','ATTENUATIONDS':'mean','ATTENUATIONUS':'me
an','MABR_DS':'mean','BADSPLICE_DET':'sum','UNBALANCED_DET':'sum
','HPN_DET':'sum', 'HPNHF_DET':'sum','IMPULSE_DET':'sum',
'MISSING_MICROFILTER_DET':'sum'}).reset_index()

print("Para el documento: "+file)
with open('performance_data.csv', 'a') as f:
      nuevo_df.to_csv(f, header=False)
```

Las operaciones consisten en una agregación por ciudad y día calculando el valor medio de las variables de rendimiento como son la potencia de señal en subida y en bajada, la atenuación o la máxima velocidad alcanzable y haciendo una suma de las distintas alarmas activadas a lo largo del día, un ejemplo de estas son las alarmas de fallo de impulso o las alarmas por cable desbalanceado.

Otro campo que aparece en el conjunto de datos resultante es ***"YEARMODA1"*** que es el valor de la fecha para el día siguiente en cada fila. Esto permitirá fácilmente cruzar la información para disponer de los valores de rendimiento para el día inmediatamente posterior.

A continuación, se ve una muestra de cómo aparecen los datos después de este procesado:

```
$ head -2 performance_data.csv

0,CITY_A,2018-10-
29,20181029,20181028,0.0,7184,1138.6230483271374,7303.084905660377,14
25.196491745283,15.589504716981095,8.917703419811044,21.6373378537736
2,12.686984080188799,26721.65386227127,97.0,2466.0,157.0,144.0,0.0,0.
0

1,CITY_A,2018-10-
29,20181029,20181028,1.0,5158,1467.7855477855478,6593.055509527755,11
19.3809030654515,16.911474730737417,10.220443247721553,27.46155758077
8926,16.344884009942106,19619.09677419355,332.0,1980.0,372.0,166.0,0.
0,0.0
```

Al agrupar los datos por ciudad y fecha, la mejora en términos de número de datos a procesar es muy significativa. El conjunto de archivos ocupaba 20G, aproximadamente 250M por cada día de datos.

```
$ wc -l Performance/performance_20181029.csv
1033526 Performance/performance_20181029.csv
$ du -bsh Performance/performance_20181029.csv
243M    Performance/performance_20181029.csv
$ du -bsh Performance/
20G     Performance/
```

Después del procesado hay únicamente un archivo csv que contiene toda la información de rendimiento de líneas:

```
$ wc -l performance_data.csv
4100 performance_data.csv

$ du -bsh performance_data.csv
925K    performance_data.csv
```

Los datos son importados en Python:

```
datos = pd.read_csv('performance_data.csv');datos

datos.columns = ['ID', 'CITY', 'INSERTION_DATE','YEARMODA',
'YEARMODA1','STABILITY','COUNT','LOOP_LENGTH','CURRENTRATEDS',
'CURRENTRATEUS','POWERDS','POWERUS','ATTENUATIONDS','ATTENUATIONUS',
'MABR_DS','BADSPLICE_DET','UNBALANCED_DET','HPN_DET',
'HPNHF_DET','IMPULSE_DET','MISSING_MICROFILTER_DET']
```

A continuación, se hace un análisis exploratorio.

El objetivo del estudio es predecir el valor de la variable ***'STABILITY'*** que determina la calidad de la conexión de los clientes. Se muestra la distribución por valor de estabilidad:

ESTABILIDAD	CUENTA	PORCENTAJE	AGRUPADO
0.0	24315150	61,40%	
1.0	9646714	24,40%	85,80%
2.0	1925796	4,90%	
3.0	3147976	7,90%	12,80%
9.0	562854	1,40%	1,40%
SUMA	39598490	100%	100%

Tabla 5-1: Distribución de líneas por cada nivel de estabilidad

Como se ve, las líneas con estabilidad óptima (0) o con ligera degradación (1) son las más numerosas con un 85,80%. Las líneas con servicio severamente degradado (2) o con servicio interrumpido debido a avería (3) suponen un 12,80%. Adicionalmente hay un 1,4% de líneas con estabilidad desconocida (9) debido a que no se han recolectado los suficientes datos durante el día.

Lo siguiente que se muestra es la distribución de los distintos valores de estabilidad por ciudad en valores absolutos.

```
sns.barplot(x="CITY", y="COUNT", hue="STABILITY", data=performance)
```

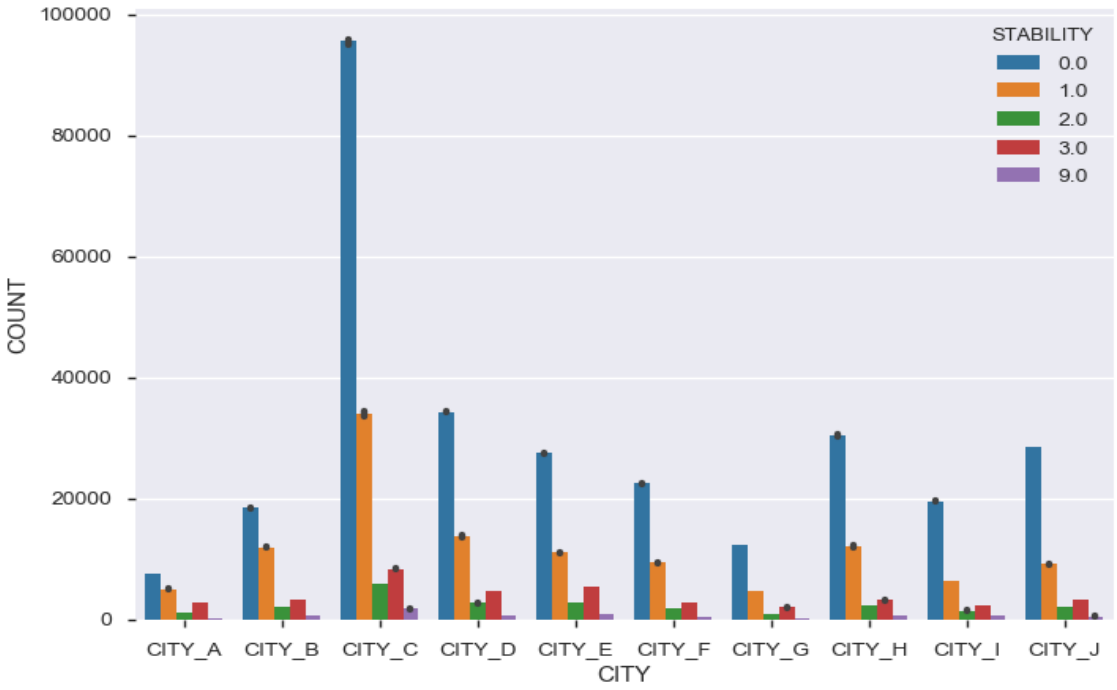

Figura 5-4: Histograma de estabilidad por ciudad

La estabilidad sigue un patrón similar en todas las ciudades, aunque desgranando por el número de clientes se observa que hay una ligera relación entre las ciudades con más clientes, previsiblemente más grandes, y la calidad de las líneas. A mayor número de clientes mayor porcentaje de líneas estables (estabilidad igual a 0) y menor porcentaje de líneas severamente degradadas (estabilidad igual a 3).

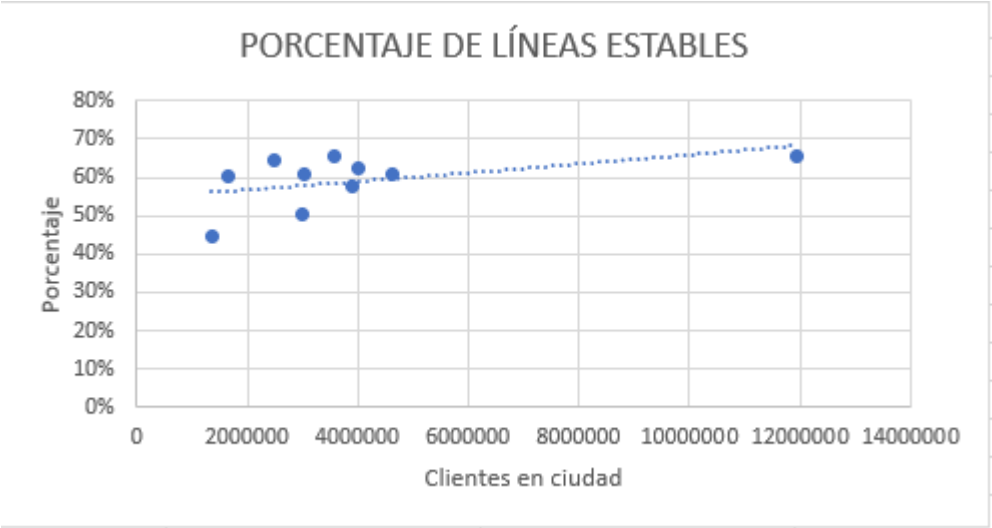

Figura 5-5: Relación entre líneas estables y clientes en la ciudad

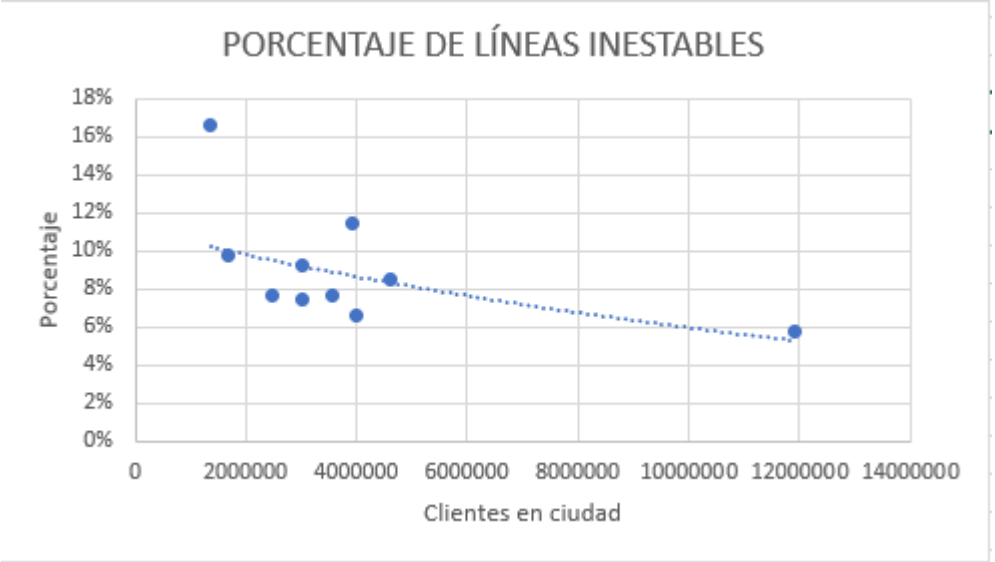

Figura 5-6: Relación entre líneas inestables y clientes en la ciudad

Se eliminan las líneas con estabilidad desconocida

```
performance =performance.loc[performance['STABILITY'] != 9]
```

Se muestra un mapa de calor de las correlaciones entre las variables que refleja la alta relación entre algunas variables.

```
correlation = performance.corr()
plt.figure(figsize=(18, 12))
heatmap = sns.heatmap(correlation, annot=False, vmin=-1,
cmap="RdBu_r")
```

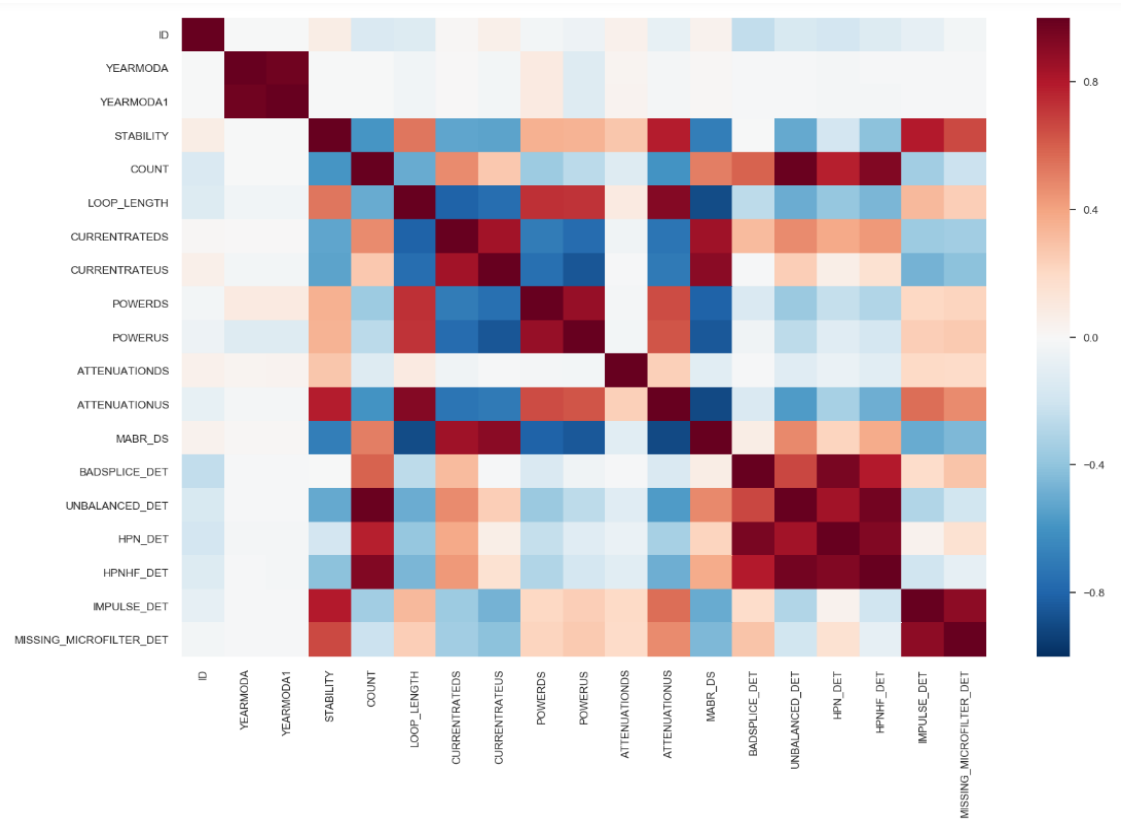

Figura 5-7: Mapa de calor con correlación entre variables

En el siguiente paso se buscan las variables que más incidencia tienen en la estabilidad. Esto ayudará en el modelo de predicción.

```
performance.corr()['STABILITY'].sort_values(ascending=False)
```

```
STABILITY                1.000000
IMPULSE_DET              0.791075
ATTENUATIONUS            0.784399
MISSING_MICROFILTER_DET  0.662347
LOOP_LENGTH              0.537565
POWERDS                  0.354180
POWERUS                  0.343925
ATTENUATIONDS            0.274784
ID                       0.076987
YEARMODA                -0.000299
YEARMODA1               -0.000307
BADSPLICE_DET           -0.007154
HPN_DET                 -0.185213
HPNHF_DET               -0.410790
UNBALANCED_DET          -0.513934
CURRENTRATEDS           -0.527067
CURRENTRATEUS           -0.537388
COUNT                   -0.590063
MABR_DS                 -0.692646
Name: STABILITY, dtype: float64
```

Como se ve, las variables más relacionadas con la estabilidad es la detección de fallo de impulso (***IMPULSE_DET***), la atenuación máxima en subida (***ATTENUATIONUS***) y la velocidad máxima alcanzable en bajada (***MABR_DS***).

Se añaden al conjunto de datos ***performance*** las variables ***ATT_US_1*** y ***MABR_DS_1*** que son los valores que toman las variables comentadas en el apartado anterior para el día siguiente.

```
dset_join = performance
dset_join = dset_join[['CITY','STABILITY','YEARMODA',
'ATTENUATIONUS','MABR_DS']]
dset_join = dset_join.rename(index=str, columns={"ATTENUATIONUS":
"ATT_US_1", "MABR_DS":"MABR_DS_1"})
performance = pd.merge(performance, dset_join,  how='inner',
left_on=['CITY','STABILITY','YEARMODA1'], right_on =
['CITY','STABILITY','YEARMODA'])
```

Estos campos se utilizarán como variable objetivo durante el proceso de modelado. Se ha desechado el uso de la variable de fallo de impulso como variable a predecir debido a que su aparición se debe principalmente a la ausencia de microfiltros en el domicilio del cliente, algo que no tiene que ver con las condiciones atmosféricas.

```
performance.corr()['IMPULSE_DET'].sort_values(ascending=False)
```

```
IMPULSE_DET                1.000000
MISSING_MICROFILTER_DET    0.891688
STABILITY                  0.791075
ATTENUATIONUS              0.561882
LOOP_LENGTH                0.320763
POWERUS                    0.248268
POWERDS                    0.208443
ATTENUATIONDS              0.197734
BADSPLICE_DET              0.184201
HPN_DET                    0.042860
YEARMODA                  -0.009043
YEARMODA1                 -0.009625
ID                        -0.086392
HPNHF_DET                 -0.201013
UNBALANCED_DET            -0.297638
COUNT                     -0.344007
CURRENTRATEDS             -0.366455
CURRENTRATEUS             -0.471018
MABR_DS                   -0.504981
Name: IMPULSE_DET, dtype: float64
```

5.3. Unión de ambos conjuntos de datos

Se unen ambas fuentes de datos haciendo coincidir los datos meteorológicos de un día con los datos de rendimiento de las líneas del día siguiente para una misma ciudad:

```
dset = pd.merge(performance, weather,  how='inner',
left_on=['CITY','YEARMODA1'], right_on = ['CITY','YEARMODA'])
```

Se dividen las muestras en 2 subgrupos: entrenamiento (aproximadamente 70% del total) y test (30%).

```
 X_train, X_test, y_train, y_test = train_test_split(dset,
dset['IMPULSE_DET'], test_size=0.3, random_state=1)
```

Se guardan los nombres de las variables climatológicas para simplificar operaciones futuras.

```
climate_vars = ['TEMP', 'DEWP', 'SLP', 'STP', 'VISIB', 'WDSP',
'MXSPD', 'GUST', 'PRCP', 'SNDP', 'maximum', 'minimum']
```

El siguiente paso seguido es un análisis bivariante entre variables de rendimiento y las variables meteorológicas. Para ello se crean subconjuntos de datos con cada una de ellas y se calcula la correlación entre las variables meterológicas y las variables de rendimiento.

En primer lugar, se analiza la correlación con la atenuación en subida.

```
subset_bivar = X_train[climate_vars+ ['ATTENUATIONUS']]
subset_bivar.corr()['ATTENUATIONUS'].sort_values(ascending=False)
ATTENUATIONUS    1.000000
minimum          0.299669
STP              0.278118
DEWP             0.269825
TEMP             0.243057
maximum          0.128352
WDSP             0.095435
VISIB            0.070325
GUST             0.063393
SLP              0.058257
MXSPD            0.034262
PRCP             0.032114
SNDP            -0.042994
Name: ATTENUATIONUS, dtype: float64
```

```
sns.jointplot(x=STP, y="ATTENUATIONUS", data=subset_bivar,
kind="reg")
```

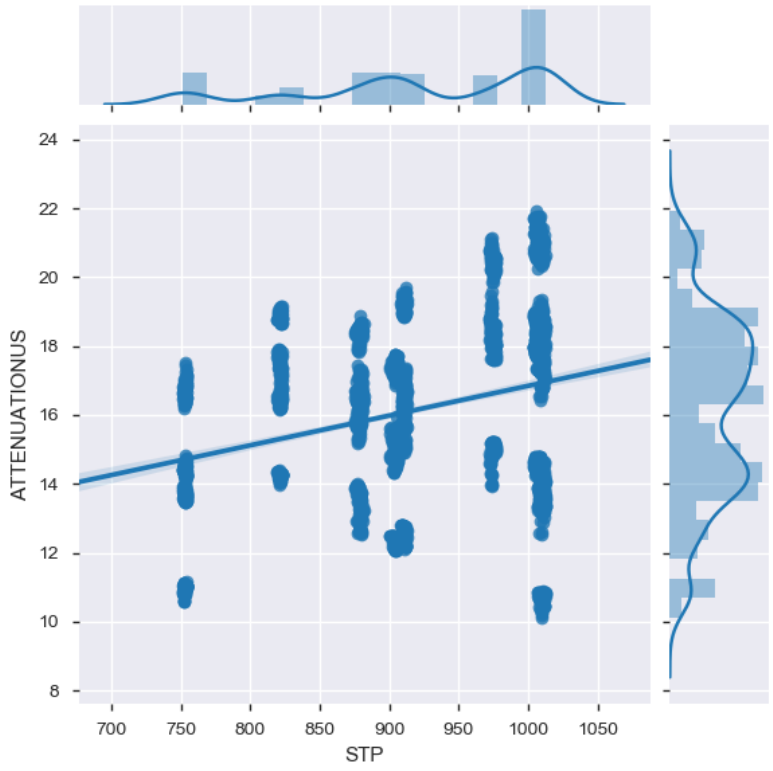

Figura 5-8: Relación entre la atenuación de señal y la presión atmosférica

Se aprecia cierta proporcionalidad entre la atenuación de señal en subida y la temperatura mínima y la presión media medida en el día, esta última representada arriba.

Se realiza la misma operación para la máxima velocidad de bajada alcanzable.

```
subset_bivar = X_train[climate_vars+ ['MABR_DS_1']]
subset_bivar.corr()['MABR_DS_1'].sort_values(ascending=False)
MABR_DS_1    1.000000
MXSPD        0.058046
SNDP         0.029609
WDSP         0.021407
maximum      0.004084
PRCP        -0.049460
TEMP        -0.051527
STP         -0.053340
GUST        -0.079813
DEWP        -0.090312
minimum     -0.107772
VISIB       -0.124657
SLP         -0.129098
Name: MABR_DS_1, dtype: float64
```

```
sns.jointplot(x="SLP", y="MABR_DS_1", data=subset_bivar, kind="reg")
```

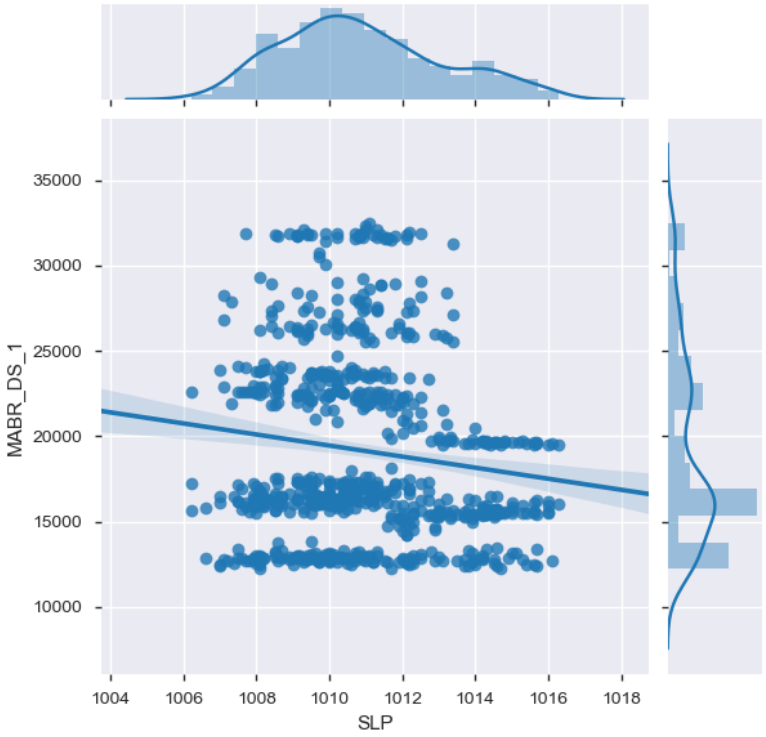

Figura 5-9: Relación entre máxima velocidad alcanzable y presión a nivel de mar

Para los indicadores de escenarios climatológicos (sol, lluvia o tormenta) se realizan algunas visualizaciones para verificar si afectan de alguna manera a la estabilidad de las líneas:

```
subset_cat = Xtrain[['STABILITY','COUNT','Cat']]

print(subset_cat.Cat.value_counts())
Sol         1711
Lluvia       780
Tormenta     280


grouped = subset_cat.groupby(['Cat','STABILITY'])['COUNT'].sum()
grouped = grouped.reset_index()
CATEGORIA    ESTABILIDAD    CUENTA    SUMA CATEGORIA    PORCENTAJE
Lluvia       0              6322397   10134245          62%
Lluvia       1              2520719   10134245          25%
Lluvia       2              491697    10134245          5%
Lluvia       3              799432    10134245          8%
Sol          0              12548471  20017327          63%
Sol          1              4833511   20017327          24%
Sol          2              991797    20017327          5%
Sol          3              1643548   20017327          8%
Tormenta     0              2239308   3670599           61%
Tormenta     1              956412    3670599           26%
Tormenta     2              178491    3670599           5%
Tormenta     3              296388    3670599           8%
```

Esta primera gráfica que se muestran los valores sin normalizar parece indicar que un clima despejado hace crecer el número de líneas estables y ligeramente degradadas.

```
sns.catplot(x="STABILITY", y="COUNT", hue="Cat", kind="point",
data=grouped);
```

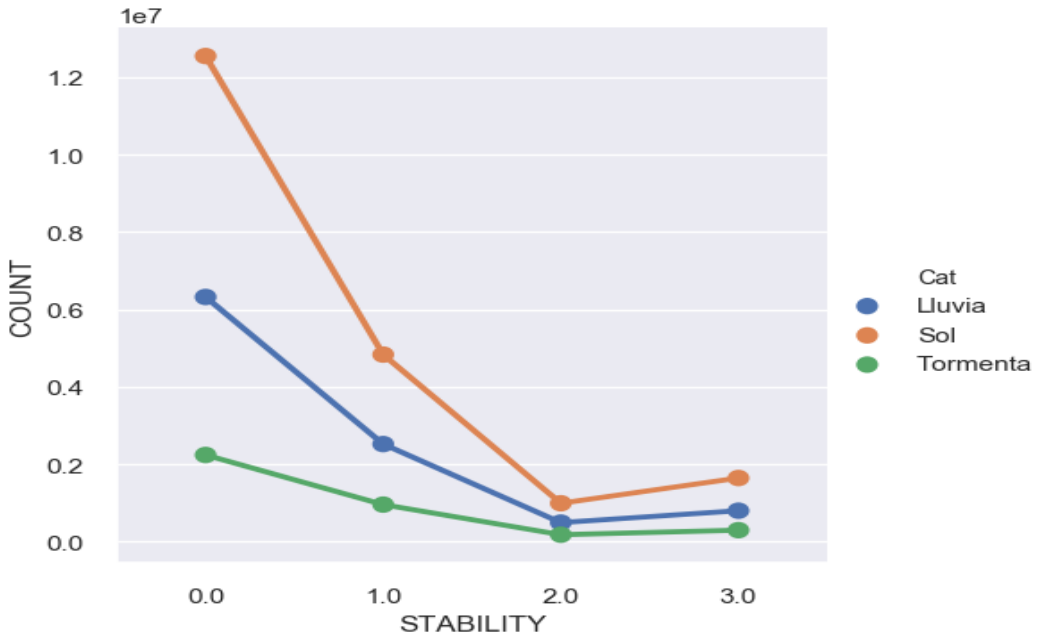

Figura 5-10: Evolución del número de líneas en cada nivel de estabilidad

Aunque, como se ve en la siguiente gráfica, teniendo en cuenta los valores porcentuales dentro de cada subgrupo, los datos representados no sirven para sacar ninguna conclusión sobre si la categoría tiene incidencia en el valor de la estabilidad.

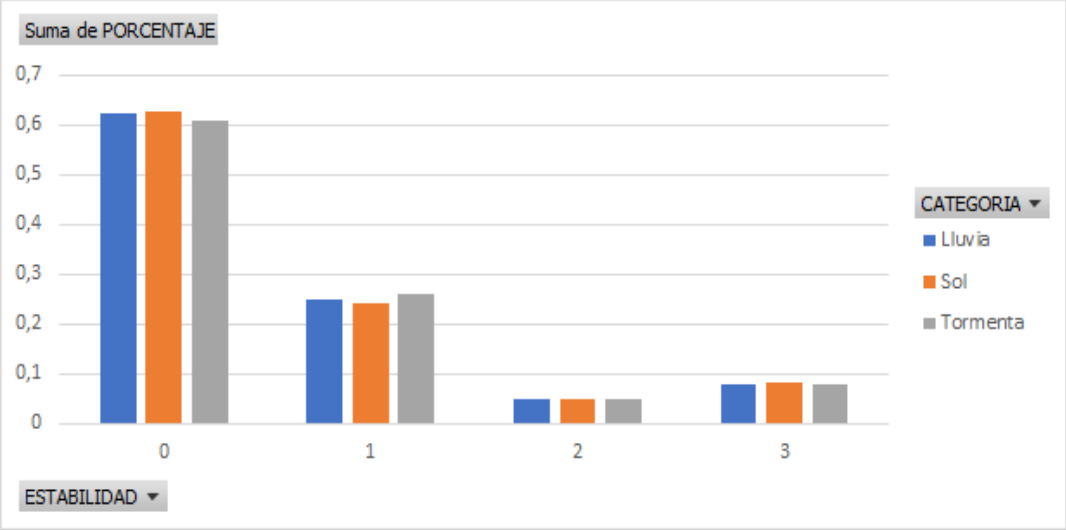

Figura 5-11: Número de líneas en cada nivel de estabilidad normalizado

Llegados a este punto se puede hacer un análisis univariante utilizando el paquete Pandas Profiling [14]

Pandas Profiling genera un informe con información descriptiva de los datos que se disponen. Se realizan distintas comprobaciones:

- Esenciales: tipo de datos, valores únicos, datos faltantes.
- Estadística descriptiva: Media, moda, desviación típica. Información por cuantiles.
- Valores más frecuentes
- Histograma
- Correlación entre variables y matrices de Spearman y Pearson.

```
import pandas_profiling as pp

profile = pp.ProfileReport(weather)
profile.to_file(outputfile="weather.html")
```

Debido a que el informe completo es bastante extenso y, en algunos casos, redundante, se mostrarán a continuación los aspectos más importantes.

Al inicio del reporte se muestra un resumen de los datos:

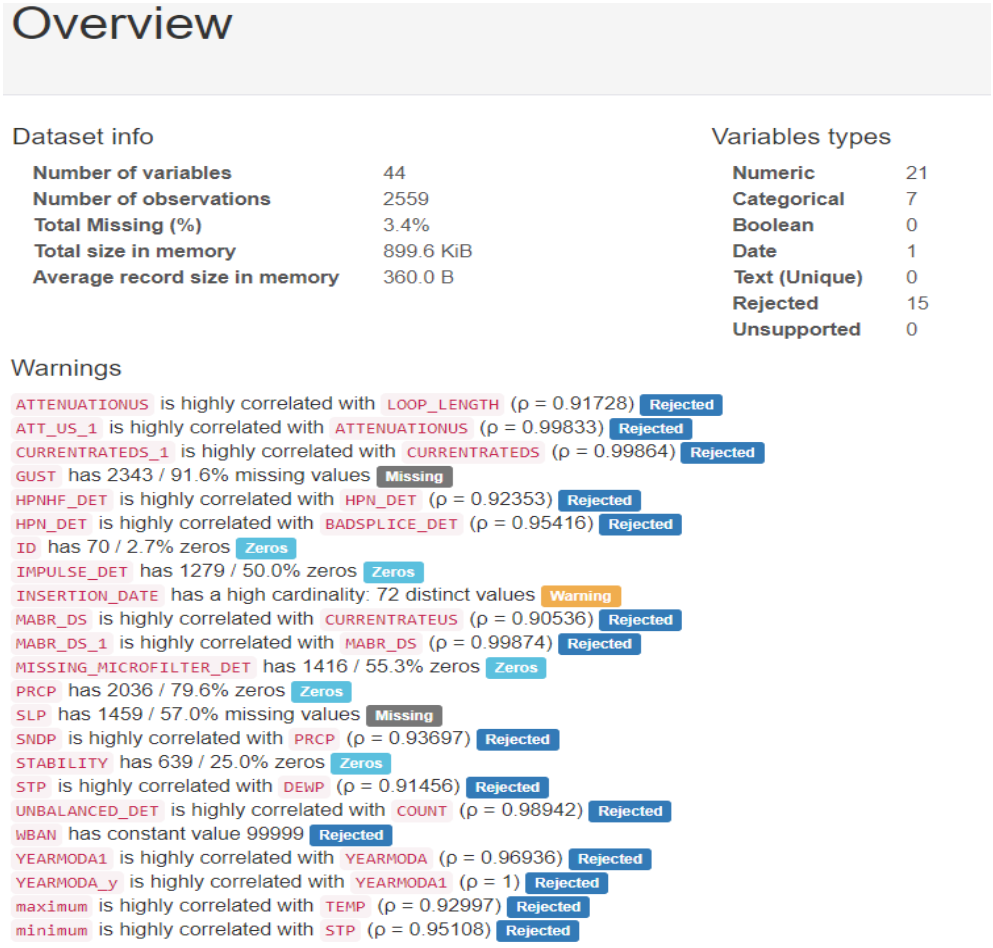

Figura 5-12: Salida del comando Pandas Profile

El informe detecta varias características del conjunto de datos, especialmente de las características de estas y su relación entre ellas.

Un ejemplo son las variables GUST (máxima ráfaga de viento detectada) y SLP (presión media a nivel del mar) que tienen un porcentaje muy alto de valores nulos, que PRCP (precipitaciones diarias) tienen un alto porcentaje de ceros, que WBAN debe ser eliminado al no aportar valor o que los campos STP (presión media en la estación), máximum (temperatura máxima) y minimum (temperatura mínima) podrían eliminarse al tener una alta correlación con DEWP (punto de rocío medio), TEMP (temperatura media) y STP respectivamente.

Aquí se muestran los resultados y las acciones a tomar en caso de ser necesarias:

- La variable GUST tiene un porcentaje de valores nulos tan alto que no puede entrar dentro del modelado.
- El campo WBAN será eliminado por tener un valor constante y no añadir información.
- Como es de esperar, las variables ATT_US_1, CURRENTRATE_1 y MABR_DS_1 tiene una alta relación con los campos que definen el valor de las mismas en el día anterior.
- Se indica que hay un alto número de ceros en las variables que acabadas en ***_DET*** que representan un valor booleano 1 ó 0 de las alarmas detectadas. No hay acciones a tomar.
- PRCP tiene un alto número de ceros, pero es esperable al tratarse de los valores de precipitación. Se comprueba viendo la distribución de valores de la variable Cat:

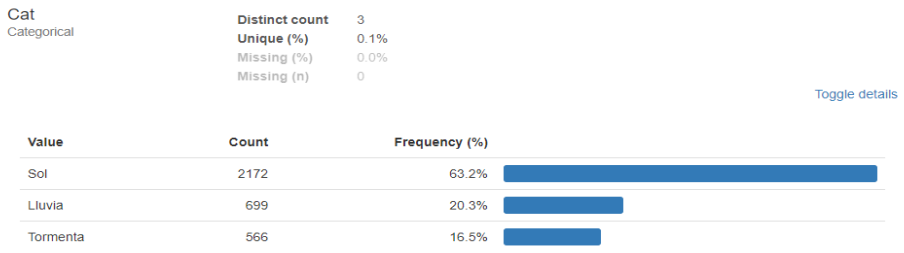

Figura 5-13: Distribución de situaciones meteorológicas

- La variable maximum se eliminará al estar altamente relacionada con TEMP
- El campo minimum está altamente correlado con STP y este, a su vez con DEWP.

```
dset[['minimum','STP','DEWP']].corr()
```

	minimum	STP	DEWP
minimum	1	0,95108	0,942666
STP	0,95108	1	0,914557
DEWP	0,942666	0,914557	1

Tabla 5-2: Matriz de correlación entre las variables minimum, STP y DEWP

```
g = sns.pairplot(dset[["minimum", "STP", "DEWP"]],
diag_kind="hist")
for ax in g.axes.flat:
    plt.setp(ax.get_xticklabels(), rotation=45)
```

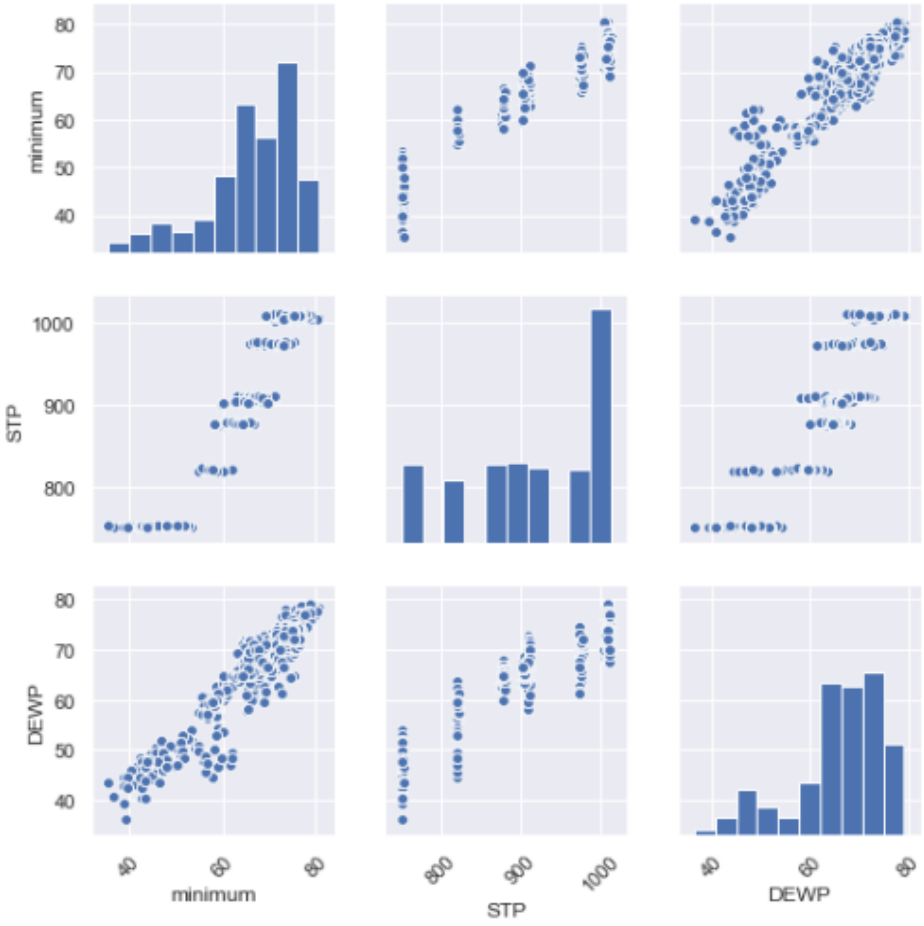

Tabla 5-3: Gráfica de las variables mininum, STP, DEWP

Tanto la matriz de correlaciones como matriz de visualizaciones pareadas muestran cómo minimun está altamente relacionada con las otras dos variables, pero no tanto STP con DEWP por lo que se decide eliminar la variable minimum.

- Para el resto de las variables con alta correlación serán eliminados a partir de valor mayor a 0.95.

- El alto número de valores nulos de la variable DEWP pueden producir pérdida de información y producir fallos en los algoritmos de modelado que se implementarán más tarde.

  Ante esta situación se pueden realizar tres acciones:

  - Dejar de tener en cuenta esa variable en el estudio posterior.
  - Imputación de datos faltantes. En función del resto de variables se puede estimar el valor que debe tener esta variables para no perder esa información. Se debe tener cautela a durante la imputación, ya que se corre el riesgo de adulterar el resultado final.
  - Eliminar filas con datos faltantes. Si el número de valores nulos es pequeño y no se quiere correr el riesgo de influir negativamente en el valor resultante, se pueden dejar de tener en cuenta esas muestras.

  Lo recomendable para este caso con un 57% de datos faltantes para la variable SLP sería la eliminación de la columna en el conjunto de datos al ser necesario generar más información de la que es provista.

  Sin embargo, se ha encontrado otra variable con muchos valores faltantes (un 4%):

```
round(dset.STP.isna().sum()/dset.STP.count(),2)
0.04
```

  A continuación, se procederá a reemplazar el 4% de valores de la variable STP que faltan por la media del valor de esa variable.

```
dset['STP'].fillna(dset['STP'].mean(), inplace = True)
```

  Como se puede observar, ya no existen valores nulos para STP:

```
print(np.isnan(dset.STP).sum())
0
```

6. Modelado

6.1. Regresión de variables de rendimiento

El primer modelo que se creará tiene por objetivo predecir el valor de las variables numéricas de la atenuación en subida y la máxima velocidad alcanzable.

Con las conclusiones del apartado anterior se crea la función “preprocessing” en la que se harán las modificaciones necesarias a los conjuntos de datos que utilizaremos:

```
 def preprocessing(df, vars_to_drop=["HPN_DET",
"INSERTION_DATE","YEARMODA", "YEARMODA1", "GUST","SNDP","SLP",
"WBAN", "UNBALANCED_DET","minimum","maximum","ID","fecha","CODE"]):

   df.drop(columns=vars_to_drop, level=None, inplace=True)
    df['STP'].fillna(df['STP'].mean(), inplace = True)
    return df
```

Luego se utiliza con los dos subconjuntos de datos para asegurar que han recibido el mismo tratamiento previo:

```
preprocessing(X_train)
preprocessing(X_test)
```

Para comprobar los modelos aplicados se va a predecir el valor de las variables elegidas utilizando el valor de esas mismas variables para el día anterior y los datos meteorológicos. De esta forma se intenta averiguar la influencia de la climatología a la hora de predecir estas variables.

Acto seguido se pondrá a prueba la efectividad de cada modelo realizando una predicción sobre el conjunto de validación (test) calculando el error medio absoluto, el error relativo medio (MAPE en inglés) y el acierto promedio, que se corresponde con el valor complementario del error relativo medio.

Se decide comenzar con un modelo sencillo como es la regresión lineal. Este consiste en la generación de una función lineal de tantas dimensiones como variables se posea asignando unos pesos específicos a cada variable que ayuden a predecir con la mayor exactitud posible el resultado final.

Modelo lineal para predecir el valor de la atenuación utilizando los datos climatológicos. Los campos de entrada serán todas las variables climatológicas, además de la atenuación y la máxima velocidad alcanzable en bajada y la salida será el valor de la atenuación para el día siguiente.

```
model = LinearRegression()
model.fit(X_train[['TEMP', 'DEWP', 'STP', 'VISIB','WDSP', 'MXSPD',
'PRCP', 'FRSHTT','ATTENUATIONUS','MABR_DS']], y_train)
```

Se realizan predicciones y se calcula el error cometido sobre el conjunto de validación:

```
predictions = model.predict(X_test[['TEMP', 'DEWP', 'STP',
'VISIB','WDSP', 'MXSPD', 'PRCP',
'FRSHTT','ATTENUATIONUS','MABR_DS']])
errors = abs(predictions - y_test)
errors_Wat = 10 ** (errors/10)
mape = 100 * (errors_Wat / y_test)
accuracy = 100 - np.mean(mape)
```

Por último, se comprueba la efectividad del modelo. Al tratarse la atenuación de una variable logarítmica, se debe convertir el error a Watios para calcular el error medio:

```
print('Mean Absolute Error:', round(10*np.log(np.mean(errors_Wat)),
2), 'dB.')
print('Mean Absolute Percentage Success:', round(accuracy, 2), '%.')
print('Mean Absolute Percentage Error', round(np.mean(mape), 2),
'%.')

Mean Absolute Error: 0.24 dB.
Mean Absolute Percentage Success: 93.49 %.
Mean Absolute Percentage Error 6.51 %.
```

Después se aplicará el mismo modelo, pero sin introducir a la entrada del algoritmo las variables meteorológicas:

```
model_bef = LinearRegression()
model_bef.fit(X_train[['ATTENUATIONUS','MABR_DS']], y_train)

predictions = model_bef.predict(X_test[['ATTENUATIONUS','MABR_DS']])
errors = abs(predictions - y_test)
errors_Wat = 10 ** (errors/10)
mape = 100 * (errors_Wat / y_test)
accuracy = 100 - np.mean(mape)


print('Mean Absolute Error:', round(10*np.log(np.mean(errors_Wat)),
2), 'dB.')
print('Mean Absolute Percentage Success:', round(accuracy, 2), '%.')
print('Mean Absolute Percentage Error', round(np.mean(mape), 2),
'%.')

Mean Absolute Error: 0.24 dB.
Mean Absolute Percentage Success: 93.49 %.
Mean Absolute Percentage Error 6.51 %.
```

Más tarde se utilizan dos algoritmos más con un resultado similar.

El primero de ellos será el algoritmo del vecino más cercano (KNN) que consiste en el cálculo del valor de salida calculando el resultado medio de las 'K' muestras más similares a la muestra a tratar.

El segundo de ellos es un algoritmo más complejo como es el del bosque aleatorios (en inglés Random Forest) que combina diversos árboles de decisión sobre las variables del conjunto de datos para inferir el resultado del modelo.

Por último, se muestra en la siguiente tabla todos los resultados obtenidos para las dos variables a predecir utilizando los distintos algoritmos.

	Atenuación					
	Regresión Lineal		KNN		Random Forest	
	Sin Meteo	Con Meteo	Sin Meteo	Con Meteo	Sin Meteo	Con Meteo
Mean Absolute Error	0.24 dB	0.24 dB	0.28 dB	0.26 dB	1.62 dB	0.97 dB
% Acierto	99.38%	99.38%	95.95%	97.52%	92.54%	93.0%
	MABR					
	Regresión Lineal		KNN		Random Forest	
	Sin Meteo	Con Meteo	Sin Meteo	Con Meteo	Sin Meteo	Con Meteo
Mean Absolute Error	161.11 kB	162.2 kB	209.25 kB	203.72 kB	186.54 kB	178.87 kB
% Acierto	99.12%	99.11%	98.86%	98.9%	99.87%	99.03%

Tabla 6-1: Resultados de modelos para variables numéricas

Aunque se profundizará más en el apartado de conclusiones, se puede observar que los datos meteorológicos utilizados no son útiles para estimar el valor de la atenuación de señal en subida ni el MABR de un día para otro, dado que la disminución del error de estimación que supone añadir dicha información es muy pequeña.

6.2. Clasificación de estabilidad

El otro modelo a implementar será un clasificador de estabilidad. Para ello se crean las siguientes variables:

- Total_count: Suma de líneas por ciudad y día.

```
performance['Total_count'] = (performance.groupby(['CITY',
'YEARMODA'])['COUNT'].transform('sum'))
```

- Perct: Porcentaje de líneas de cada nivel de estabilidad por ciudad y día.

```
performance['Perct'] =
performance['COUNT']/performance['Total_count']
```

- Mean: Porcentaje medio de líneas en cada nivel de estabilidad por ciudad y día

```
stable= performance.loc[performance['STABILITY'].isin([0,1])]
stable['MEAN'] =
(stable.groupby(['CITY'])['Perct'].transform('mean'))
```

- Summary: Variable categórica; 1 si el día presenta mayor porcentaje de líneas estables que la media, 0 si no.

```
stable['summary'] = np.where(stable["Perct"] > stable['MEAN'] ,
1, 0)
```

- Summary_1: Valor del campo Summary para el día siguiente. Variable a predecir.

```
dset_join = stable
dset_join = dset_join[['CITY','YEARMODA1','summary']]
dset_join = dset_join.rename(index=str, columns={"summary":
"summary_1"})
stable = pd.merge(stable, dset_join,  how='inner',
left_on=['CITY','YEARMODA'], right_on = ['CITY','YEARMODA1'])
```

Se realiza una pequeña exploración de las variables nuevas:

```
stable['summary_1'].value_counts()
1    387
0    363
```

```
stable[['CITY','Perct']].groupby('CITY').describe().reset_index
```

CITY	count	mean	std	min	25%	50%	75%	max
CITY_A	82,00	0,7488	0,0240	0,5426	0,7456	0,7516	0,7561	0,7667
CITY_B	82,00	0,8301	0,0047	0,8160	0,8279	0,8306	0,8327	0,8418
CITY_C	82,00	0,8897	0,0039	0,8785	0,8872	0,8900	0,8922	0,8983
CITY_D	82,00	0,8535	0,0037	0,8456	0,8513	0,8528	0,8549	0,8654
CITY_E	82,00	0,8091	0,0072	0,7901	0,8039	0,8103	0,8140	0,8225
CITY_F	82,00	0,8618	0,0077	0,8412	0,8581	0,8622	0,8658	0,8786
CITY_G	82,00	0,8395	0,0054	0,8290	0,8365	0,8392	0,8432	0,8563
CITY_H	82,00	0,8736	0,0051	0,8658	0,8700	0,8723	0,8758	0,8868
CITY_I	82,00	0,8540	0,0074	0,8309	0,8495	0,8562	0,8594	0,8650
CITY_J	82,00	0,8629	0,0048	0,8517	0,8599	0,8624	0,8671	0,8720

Tabla 6-2: Porcentaje de líneas estables por ciudad

El porcentaje de líneas estables parece tener relación con el número de clientes en la ciudad:

```
sns.regplot(x="Total_count", y="Perct", data=stable, logistic=True)
```

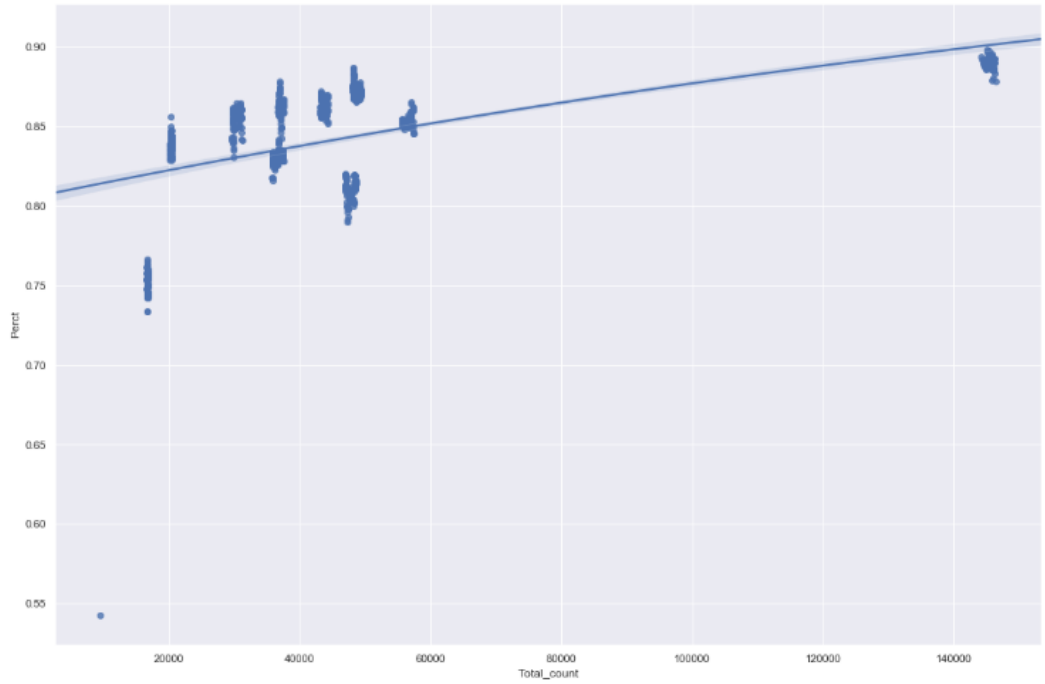

Figura 6-1: Porcentaje de líneas estables por ciudad

Con esto se muestra que las poblaciones más grandes disponen de mejor porcentaje de líneas estables y por tanto un mejor servicio medio por cliente.

El siguiente paso es crear variables “dummy” a partir de la variable ciudad para convertir en numérica la variable categórica:

```
cat_list = pd.get_dummies(stable['CITY'], prefix='C')
data1=stable.join(cat_list)
stable=data1

data_vars=stable.columns.values.tolist()
to_keep=[i for i in data_vars if i != 'CITY']
```

Se ve cómo ya no es necesaria la utilización de una variable no numérica como "***CITY"***:

```
print(dset[['CITY','C_CITY_A', 'C_CITY_B', 'C_CITY_C','C_CITY_D',
'C_CITY_E', 'C_CITY_F', 'C_CITY_G', 'C_CITY_H', 'C_CITY_I',
'C_CITY_J']].head())
```

CITY	C_CITY_A	C_CITY_B	C_CITY_C	C_CITY_D	C_CITY_E	C_CITY_F	C_CITY_G	C_CITY_H	C_CITY_I	C_CITY_J
CITY_B	0	1	0	0	0	0	0	0	0	0
CITY_C	0	0	1	0	0	0	0	0	0	0
CITY_D	0	0	0	1	0	0	0	0	0	0
CITY_E	0	0	0	0	1	0	0	0	0	0
CITY_G	0	0	0	0	0	0	1	0	0	0

Tabla 6-3: Muestra de variables "dummy" del campo CITY

Se parte el conjunto de datos en entrenamiento, validación y test.

```
 X_train, X_test, y_train, y_test = train_test_split(dset,
dset['summary_1'], test_size=0.3, random_state=1)
```

Al igual que para los modelos de regresión se realiza un modelado teniendo en cuenta las variables climáticas y sin ellas.

Sin variables climáticas

```
def preprocessing_wo_weather(df):

    df = df[['summary','C_CITY_A', 'C_CITY_B', 'C_CITY_C','C_CITY_D',
'C_CITY_E', 'C_CITY_F', 'C_CITY_G', 'C_CITY_H',
'C_CITY_I','C_CITY_J']]

    return df

X_train_ww = preprocessing_wo_weather(X_train)
X_test_ww = preprocessing_wo_weather(X_test)
```

Se elige el algoritmo de regresión logística al ser uno de los algoritmos más potentes a la vez que sencillo de comprender.

```
from sklearn.linear_model import LogisticRegression
logisticRegr_ww = LogisticRegression()
logisticRegr_ww.fit(X_train_ww, y_train)
predictions = logisticRegr_ww.predict(X_test_ww)
score_ww = logisticRegr_ww.score(X_test_ww,y_test)
print(score_ww)

0.8769230769230769
```

Se muestra la matriz de confusión que indica cómo han sido las decisiones tomadas por el modelo comparadas con el valor real para los dos valores posibles de salida.

```
from sklearn import metrics
cm_ww = metrics.confusion_matrix(y_test, predictions)

plt.figure(figsize=(9,9))
sns.heatmap(cm_ww, annot=True, fmt=".3f", linewidths=.5, square =
True, cmap = 'Blues_r');
plt.ylabel('Valor real');
plt.xlabel('Valor predicho');
all_sample_title = 'Puntuación de acierto: {0}'.format(score)
plt.title(all_sample_title, size = 15);
```

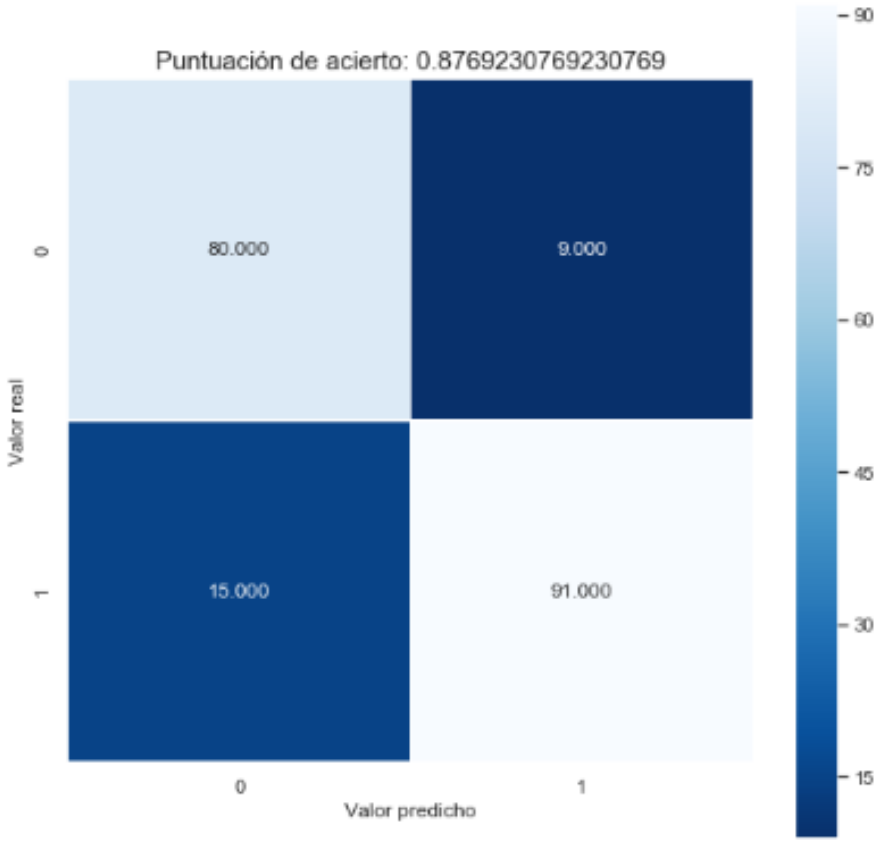

Figura 6-2: Matriz de confusión de estabilidad sin datos climatológicos

Con variable climáticas

```
def preprocessing(df):

     df = df[['TEMP', 'DEWP', 'STP', 'VISIB', 'WDSP', 'MXSPD',
'PRCP','summary','C_CITY_A', 'C_CITY_B', 'C_CITY_C','C_CITY_D',
'C_CITY_E', 'C_CITY_F', 'C_CITY_G', 'C_CITY_H',
'C_CITY_I','C_CITY_J']]

    df['STP'].fillna(df['STP'].mean(), inplace = True)

    return df

X_train = preprocessing(X_train)
X_test = preprocessing(X_test)

from sklearn.linear_model import LogisticRegression
logisticRegr = LogisticRegression()
logisticRegr.fit(X_train, y_train)
predictions = logisticRegr.predict(X_test)
score = logisticRegr.score(X_test,y_test)
print(score)

0.8769230769230769
```

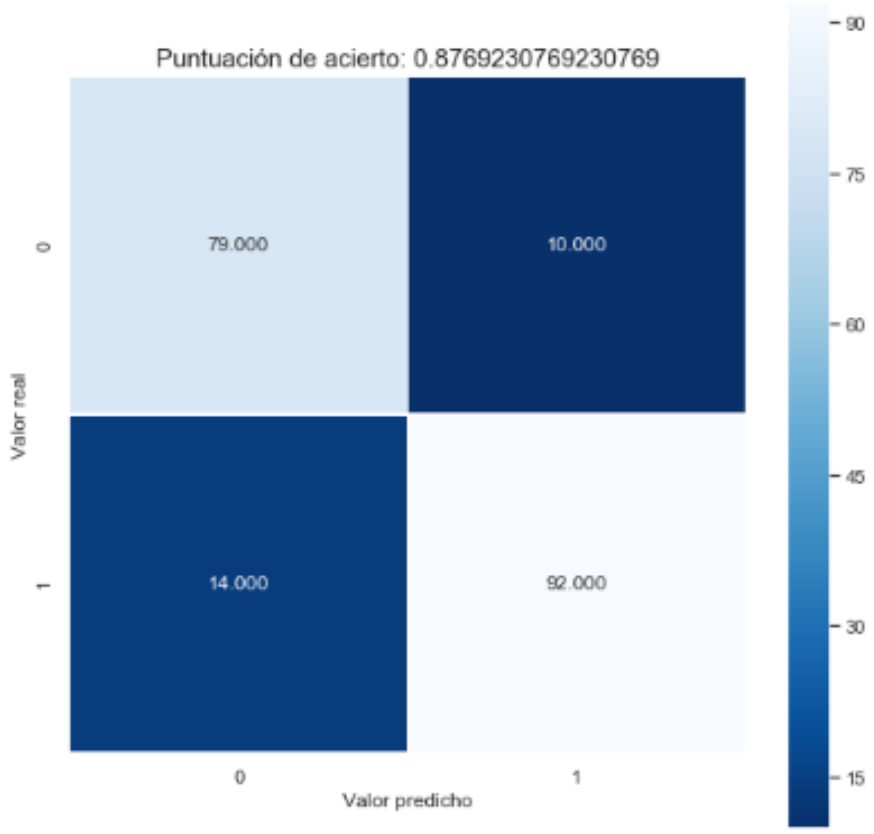

Figura 6-3: Matriz de confusión de estabilidad con datos climatológicos

Ambos modelos obtienen una puntuación de acierto idéntica y una matriz de confusión muy similar.

Pese a eso, se observa que las variables climatológicas sí tienen cierto peso en la decisión, aunque el campo con mayor relevancia sea la misma variable para el día anterior:

```
X_train.columns
Index(['TEMP', 'DEWP', 'STP', 'VISIB', 'WDSP', 'MXSPD', 'PRCP',
'summary',_CITY_A', 'C_CITY_B', 'C_CITY_C', 'C_CITY_D', C_CITY_E',
'C_CITY_F', _CITY_G', 'C_CITY_H', 'C_CITY_I', 'C_CITY_J'],
dtype='object')

logisticRegr.coef

array([[-1.07055752e-01,3.79542652e-02, 5.54823302e-03,
        -4.50751210e-02,2.44015172e-03,-3.04880533e-02,
        -1.16410640e+00,2.96618161e+00, 2.48243859e-01,
        -6.35543765e-01,-2.15180515e-01,-4.65195467e-01,
         5.53745528e-01,0.00000000e+00,-7.32218164e-02,
         6.86614785e-02,4.33563872e-01,1.52051340e-01]])
```

Por esto, se puede deducir que el conocimiento de los sucesos meteorológicos añade valor a la predicción de la estabilidad en las líneas, pero no ha sido posible demostrarlo con el conjunto de datos disponible.

7. Cronograma y presupuesto

7.1. Cronograma

En este apartado se detalla el cronograma que se ha seguido en la realización del proyecto:

N.º	Fecha de inicio	Fecha de finalización	Actividad	Duración en jornadas
1	02/03/2019	03/03/2019	Redes ADSL	2
2	02/03/2019	03/03/2019	Aprendizaje máquina	2
3	03/03/2019	06/03/2019	Modelado de datos	4
4	07/03/2019	07/03/2019	Contraste de hipótesis	1
5	08/03/2019	12/03/2019	Establecimiento de caso de uso	5
6	13/03/2019	18/03/2019	Inventario de datos de rendimiento disponibles	6
7	19/03/2019	23/03/2019	Búsqueda de datos meteorológicos	5
8	24/03/2019	07/04/2019	Obtención y procesado de datos de rendimiento	15
9	08/04/2019	22/04/2019	Obtención y procesado de datos meteorológicos	15
10	23/04/2019	02/05/2019	Aplicación de diferentes modelos al conjunto de datos	10
11	03/05/2019	12/05/2019	Conclusiones y siguientes pasos	10
12	13/05/2019	27/05/2019	Redacción de la memoria	15

Tabla 7-1: Cronograma del proyecto

A continuación, se muestra en diagrama de Gantt:

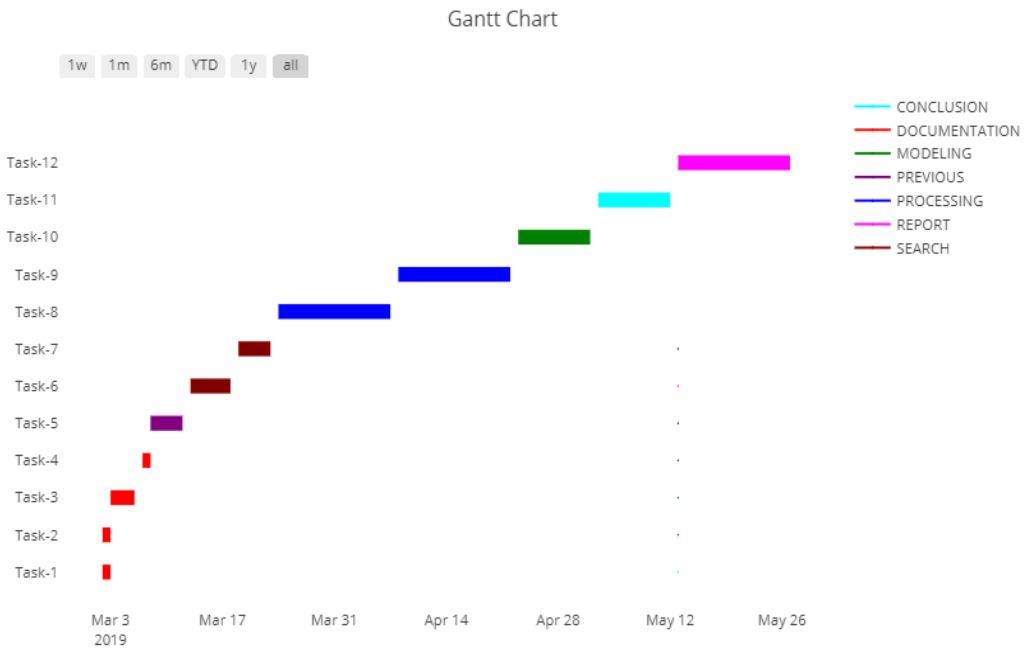

Figura 7-1: Diagrama de Gantt

7.2. Presupuesto

En este apartado se muestra el presupuesto necesario para llevar a cabo el proyecto. Se incluirá tanto el coste de personal como el coste de material.

En primer lugar, como se comentaba en el apartado del cronograma son necesarias 90 jornadas de trabajo durante 18 semanas. Se estima además una dedicación de 4 horas diarias al proyecto.

Además, hay que tener en cuenta las horas dedicadas por el tutor para el guiado y corrección de errores que han supuesto dos horas semanales en toda la duración del proyecto.

Personal	**Tiempo de trabajo (horas)**	**Coste por hora (€/h)**	**Coste (€)**
Tutor	36	80	2880
Alumno	360	40	14400
		TOTAL	17280

Tabla 7-2: Distribución de costes personales

A los costes personales es necesario sumar los costes indirectos, correspondientes al 10% de estos: 1.728 €.

Por otro lado, es necesario tener en cuenta los costes asociados al material utilizado. Dado que Python como las librerías utilizadas son de código libre, no se ha producido costes asociados a software. Sin embargo, ha sido necesaria la adquisición de un ordenador portátil con la suficiente capacidad para realizar el procesado de gran cantidad de datos y un disco duro SSD para optimizar el funcionamiento.

Para calcular el coste material es necesario tener en cuenta las amortizaciones. Se aplica el coeficiente de amortización establecido por la Agencia Tributaria [1]. La fórmula para calcular el coste sería la siguiente:

Coste = Coef. amortización x Nº meses utilización / 12 x Precio del material

Material	**Precio unidad (€)**	**Nº meses de utilización**	**Coef. Amortización**	**Coste (€)**
Ordenador portátil	600	4	0,26	52,00
Disco duro SSD	150	4	0,26	13,00
			TOTAL	65,00

Tabla 7-3: Distribución de costes materiales

Presupuesto	
Costes materiales	65
Costes personales	17280
Costes indirectos	1728
Total	19073
IVA (21%)	4005,33
TOTAL (IVA incluido)	23078,33

Tabla 7-4: Presupuesto del proyecto

Haciendo la suma de los costes materiales y personales el coste total asciende a 23.078.33 €.

8. Conclusiones

Para analizar las conclusiones de este trabajo es necesario mirar los objetivos planteados. Estos objetivos se pueden dividir en dos planos:

- Plano de negocio: La motivación de este proyecto era realizar una exploración sobre cómo afecta la climatología al rendimiento de las líneas y cómo sería el proceso que habría que seguir para intentar obtener resultados medibles de este fenómeno.

  Si bien es cierto que la conclusión que se puede deducir de este proyecto es que no hay ninguna relación medible entre los datos meteorológicos y la calidad de las comunicaciones, sí se puede afirmar que se han sentado unas bases para un proyecto de análisis de datos más profundo.

  Siguiendo los pasos del diagrama del proceso CRISP-DM, ahora mismo nos encontramos tras el proceso de evaluación del modelo volviendo al punto del conocimiento del negocio y de los datos. Dado que el proceso actual no obtiene el resultado esperado se debe redefinir el caso de negocio o bien volver a realizar un proceso de búsqueda, obtención y tratamiento de nuevos datos que sean de utilidad para la creación de nuevos modelos.

- Plano académico: A pesar de que los resultados de los análisis no han resultado como se esperaba, desde el punto de vista educativo este proyecto ha sido un éxito.

  La realización de un proyecto de análisis de datos ha requerido la preparación de un caso de uso, que lleva consigo la comprensión del área de negocio a tratar. Además, la obtención de datos y comprensión de la información disponible ha necesitado de la redefinición o perfilamiento del proyecto.

  Desde el punto de vista técnico este trabajo ha servido para comprender y coger soltura las herramientas más utilizadas en este tipo de proyectos, en especial el lenguaje SQL y el lenguaje Python con sus librerías pandas, seaborn o scikit-learn.

9. Siguientes pasos

Como se comenta en el apartado anterior, el final de este proyecto no supone el final del análisis. Existen múltiples puntos de acción con los que obtener unos datos más adecuados a este análisis, además de una posible redefinición del caso.

En primer lugar, lo primero en lo que se debería poner un esfuerzo sería aumentar la cantidad de datos de los que se dispone. El conjunto de datos actual contiene datos de tres meses. Sería interesante estudiar cómo se comportan las líneas en diferentes estaciones además que, aunque una mayor cantidad de datos no implica necesariamente mayor cantidad de información útil, es muy probable que los resultados que se obtengan sean más precisos.

Otro punto que ayudaría a obtener un resultado más preciso sería no agrupar los datos por ciudad y día ya que se reducen el número de muestras de manera considerable. Algunas ciudades de las que se estudian tienen una extensión muy amplia por lo que un fenómeno meteorológico puede no afectar de igual forma a distintas partes de la ciudad. Puesto que tratar individualmente a cada abonado puede ser inabarcable por la cantidad de datos que habría que procesar podría hacerse un estudio por código postal.

Tanto la información de rendimiento como meteorológica representan unos valores medios diarios, por lo que un caso como un corte de servicio momentáneo provocado por una tormenta sería difícil de detectar. Una información detallada por horas podría ser más útil para el estudio.

Sería también de mucha utilidad añadir la información de llamadas al centro de atención y las salidas de técnico al domicilio del cliente. Esto, además de ser un indicador inequívoco de mala calidad en la conexión es un parámetro añaden una variable crucial al estudio, el coste.

Dado que cada llamada o despacho a terreno tienen asociado un coste, se presenta un escenario nuevo y con muchos puntos de trabajo. ¿Es más costoso arreglar una línea en buen estado o no actuar con un cliente que experimenta una conexión inestable? Asimismo, se presenta un caso de negocio muy claro que presentar a un operador, ¿cuánto aumenta los costes la climatología?

Bibliografía

[1] Agencia Tributaria, «Estimación directa simplificada,» [En línea]. Available: https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresarios_individuales_y_profesionales/Rendimientos_de_actividades_economicas_en_el_IRPF/Regimenes_para_determinar_el_rendimiento_de_las_actividades_economicas/Es.

[2] Temas tecnológicos, [En línea]. Available: https://www.temastecnologicos.com/redes-fijas/. [Último acceso: 2019].

[3] D. Conway, «The Data Science Venn diagram,» 30 Septiembre 2010. [En línea]. Available: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram. [Último acceso: 2019].

[4] A. J. Kelleher B. Mac Namee, de *Fundamentals of Machine Learning for Predictive Data Analytics.* , MIT Press, 2015.

[5] G. Grolemund y H. Wickham, «R for Data Science,» 2017. [En línea]. Available: https://r4ds.had.co.nz/model-intro.html. [Último acceso: 2019].

[6] D. Soni, «Supervised vs. Unsupervised Learning,» 2018. [En línea]. Available: https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d. [Último acceso: 2019].

[7] Parlamento Europeo y consejo de la Unión Europea, «REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO,» 27 Abril 2016. [En línea]. Available: https://eur-lex.europa.eu/legal-content/ES/TXT/HTML/?uri=CELEX:32016R0679&from=ES.

[8] «Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.,» 6 Diciembre 2018. [En línea]. Available: https://boe.es/boe/dias/2018/12/06/pdfs/BOE-A-2018-16673.pdf. [Último acceso: 2019].

[9] Agencia Española de Protección de Datos, «Responsabilidad proactiva,» [En línea]. Available: https://www.aepd.es/reglamento/cumplimiento/principio-responsabilidad-proactiva.html. [Último acceso: 2019].

[10] Grupo Ático34, «Claves de la LOPDGDD,» 27 Noviembre 2018. [En línea]. Available: https://protecciondatos-lopd.com/empresas/claves-de-la-lopd-gdd/. [Último acceso: 2019].

[11] Grupo Ático34, «Consentimiento para el tratamiento de los datos personales según el RGPD,» 17 Enero 2018. [En línea]. Available: https://protecciondatos-lopd.com/empresas/consentimiento-rgpd/.

[12] National Centers for Enviromental Information, «National Centers for Enviromental Information,» [En línea]. Available: https://www.ncdc.noaa.gov/.

[13] National Climatic Data Center (NCDC), [En línea]. Available: https://www7.ncdc.noaa.gov/CDO/GSOD_DESC.txt. [Último acceso: 2019].

[14] Pandas profiling, «Create HTML profiling reports from pandas DataFrame objects,» [En línea]. Available: https://github.com/pandas-profiling/pandas-profiling. [Último acceso: 2019].